EXE

OCTOBER 1991 VOL 6 ISSUE 5

£3.20

The Software Developers' Magazine

Inside the dBASE dBATE

Open house for Open Systems.
Extra UNIX coverage this month.

Raw I/O causes terminal anxiety -
Ben Thompson sorts out getchar ( ) problems.

All roads lead to System V Release 4.
What has this UNIX panacea to offer?

Microsoft's new QuickC hits the streets.

Graphics galore: 3D hidden lines,
Windows libraries & loading bitmaps.

Vol 6 Issue 5

.EXE

October 1991

General
.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 044282 4501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions
.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print.

Editorial
Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising
If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation
The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Issue theme: UNIX

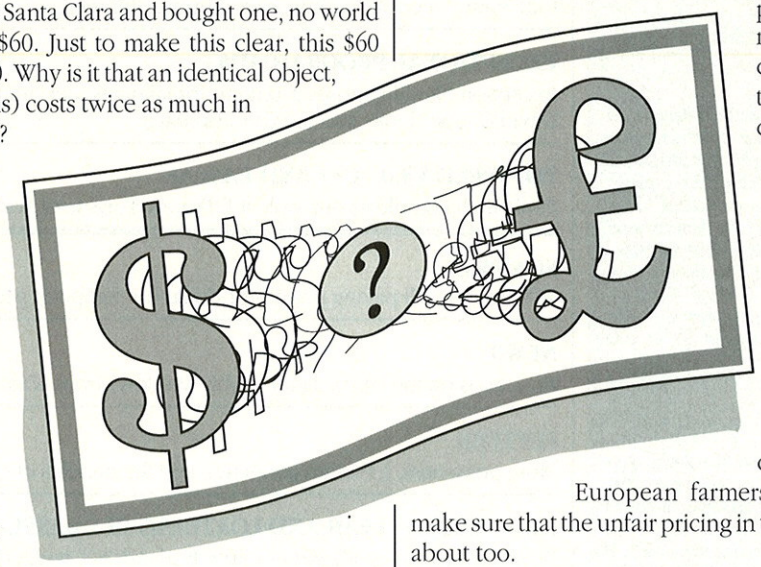
The Third Side is on holiday, and will return next month.

Money makes the world go round

Peter Collinson has been bitten by the pound-for-dollar price increase that appears over the Atlantic.

I was recently packing up to attend a meeting in Silicon Valley when the NiCad battery on my Toshiba laptop died. I started ringing around all the mail order companies to find a replacement or two and found that there was a world shortage. I eventually found a company with one on the shelf and ordered it. £50, but I needed it. Right?

I landed in the USA and thought that it would be good to have another battery, doubling the working time of the machine. I arrived at Fry's Electronics in Santa Clara and bought one, no world shortage, no problem, but $60. Just to make this clear, this $60 translates into around £36.50. Why is it that an identical object, made in Japan (or so it seems) costs twice as much in the UK as it does in the USA?



You can't just blame Toshiba. Sun Microsystems has just placed all its manual pages and documents on a CD. The idea is that it will replace all the paper that they used to generate per customer with one single piece of plastic. This will save it the cost of about 2 foot of binders sitting on shelves gathering dust. The plastic is better than paper since it contains hyperlinks, making navigation easier. This CD will be produced in thousands and possibly tens of thousands. The basic media cost will be perhaps a couple of pounds. In the USA, a Sun customer paying the list price for the CD will be $425 out of pocket. It will cost the unsuspecting UK customer twice that: £400. Why?

Look at the Apple Mac story. My US friends are astounded at how few Macs there are in the UK. The Mac has been seen as a cheap machine for years in the US. Cheap enough for home use. Cheap enough to dump in an office next to a printer and used when needed as a tool for document preparation. Cheap enough to be used everywhere in education. You can find stores on the street corner in the smallest of towns that will take your Mac floppy disk and print the words using a high quality laser printer.

This is all very rare here and it's easy to see why. We have always thought that the Mac is an expensive machine and so people haven't bought them. Since they are not so widespread there is only a tiny support industry providing add-on service to Mac users.

These stories on not uncommon, and I really am not singling out these companies as being bad. Dollar for pound pricing seems to be the order of the day. It's interesting that selling software at close to US prices is being used as a marketing ploy by several PC software vendors. To me this shows that pricing is not a matter of tariffs, extra costs or even rational decision making. It's what the market will bear.

When challenged about over-pricing (whoops, I meant high pricing) vendors will trot out many excuses. First, it's shipping charges, excise duty, and VAT that put the prices up. I am unconvinced by these arguments. This would put the price up, but not double it.

One claim I have heard is that the US vendors just automatically have higher prices for non-domestic customers. If this is true, maybe we should be loudly shouting 'unfair' in the Government's hearing-aid. The US is complaining that subsidy for European farmers is unfair competition. Let's make sure that the unfair pricing in the computer industry is known about too.

Another excuse is that the companies have to run local support departments and have to charge extra to pay for them. I really don't buy this either. People are cheaper here and the component of the price that pays for support must stay here to fund the local staff. I cannot see this doubling the price.

Why complain? Let's just pay up or shut up. Well, in the long run, all this is not just hurting our pockets. It hurts the UK industry as a whole. The entire computing infrastructure of the country costs twice as much as it should. This is stunting its growth and restricting all our opportunities. We are supposed to be good at software, but are being held back by the high cost of all sorts of pieces of the computing environment that people in the US take for granted. This makes me cross.

EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

GREY MATTER

Prigg Meadow, Ashburton, Devon TQ13 7DF

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe.
Please check prices at time of order as ads are prepared some weeks before publication.
This page lists some of our products - call us for a complete price list and details of discount structure.
ORDER BY PHONE WITH YOUR CREDIT CARD.

TEL: (0364) 53499
FAX: (0364) 53071

C++ COMPILERS

Topspeed C++ Std	OS/2	£110
Topspeed C++ Prof	OS/2	£165
Zortech C++ Dev v3	DOS&OS/2	£360
Zortech C++ Sci&Eng	DOS&OS/2	£540
Borland C++	PC-DOS	£206
Guidelines C++	PC-DOS	£370
Topspeed C++ Std	PC-DOS	£110
Topspeed C++ Prof	PC-DOS	£165
Turbo C++	PC-DOS	£ 50
Zortech C++ v3	DOS&Windows	£225

LIBRARIES

C++/Views (ZC++)	WIN	£270
Zinc Interface Lib (	C++)PC-DOS	£155
Zinc Interface Lib	WIN&DOS	£230
Zortech C++ Database	PC-DOS	£180
Win++	WIN	£160
Codebase++	WIN&DOS	£195

C LIBRARIES & TOOLS

COMMS

C Asynch Manager V3s'ce	PC-DOS	£135
Essential Comms(MS&TC)	PC-DOS	£180
Greenleaf Comms(source)	PC-DOS	£195
Silvercomm C Asynch	PC-DOS	£140

DATABASE/FILEHANDLING

Btrieve for OS/2	OS/2	£420
Btrieve for Windows	WIN	£420
Btrieve v5.1	MS-DOS	£420
XQL	OS/2,PC-DOS	£560
Xtrieve Plus	OS/2,PC-DOS	£420
C/Database T'lchest sce	PC-DOS	£ 45
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£170
C-tree Plus	PC-DOS	£360
Essential Btree (s'ce)	PC-DOS	£110
Codebase 4.21		£195
CQL s'ce	PC-DOS	£270
Faircom Toolbox Prof v2	PC-DOS	£680
R-tree	MS-DOS	£160
db-VISTA DBMS Object	PC-DOS	£435
SftFcus Btree&Isam	(sce any C)	£ 80

GRAPHICS

Essential Graphics v4	PC-DOS	£220
GFX Graphics	PC-DOS	£ 80
GraphiC v5 (MS)	PC-DOS	£245
graphics MENU/native	PC-DOS	£395
GSS Kernel system	OS2,PC-DOS	£425
GSS Graph Dev T'kit	OS2,PC-DOS	£425
GX Graphics	PC-DOS	£105
HALO v3 (MSC5,TC)	PC-DOS	£215
HALO Professional (C)	PC-DOS	£335
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£190
MetaWINDOW v3.7c	PC-DOS	£165
MetaWINDOW/Plus	PC-DOS	£220
TurboWINDOW/C (Turbo)	PC-DOS	£ 90
QuickWINDOW/C (QC)	PC-DOS	£ 90

SCREEN & WINDOWS

C-Scape +Look&Feel	OS/2&PC-DOS	£390
Vitamin C for OS/2	OS/2&DOS	£240
C-Worthy w CW Architect	MS-DOS	£410
Formation Desktop	PC-DOS	£140
Greenleaf Datawindow	PC-DOS	£215
Greenleaf Makeform	PC-DOS	£ 75
Mewel	PC-DOS	£250
Panel Plus II	(source) PC-DOS	£260
Vermont Views (MSC,TC)	PC-DOS	£345
Vitamin C	(source) PC-DOS	£270

GENERAL FUNCTIONS

C/Utilities T'chest sce	PC-DOS	£ 45
C Tools Plus/6.0 (MS)	PC-DOS	£105
CodeRunner	PC-DOS	£140
Entelekon Funct(source)	PC-DOS	£105
Essential C Utils v5	PC-DOS	£140
G'leaf Fin'cial Mathlib	PC-DOS	£215
G'leaf Super Functions	PC-DOS	£160
Greenleaf Functions s'ce	PC-DOS	£125
Heap Checker	PC-DOS	£ 30
Lattice C Comp Kit+s'ce	PC-DOS	£160
PDK£1	PC-DOS	£ 70
Power Search	PC-DOS	£ 99
Resident-C (MS,L)(s'ce)	PC-DOS	£135
Silverware C EMM Lib	PC-DOS	£105
Heap Expander(MS&TC)	PC-DOS	£ 65
Turbo C Tools v2	PC-DOS	£ 95
WKS Library	PC-DOS	£160

MATHEMATICAL LIBRARIES

C/Math Toolchest&Grafix	PC-DOS	£ 45
C Lang Sci Lib v2	ANY C	£160
Mathpak 87 (L,MS)	MS-DOS	£ 85
Science & Eng Tools	MS-DOS	£ 79

TOOLS & SUNDRIES

PC-Lint	OS/2&DOS	£ 85
Clear+ for C	PC-DOS	£140
Databoss (C)	PC-DOS	£320
Lattice Comp Companion	MS-DOS	£ 70

REAL TIME & MULTI-TASKING TOOLS

Aspen Multi-C (MSC,TC)	PC-DOS	£ 90
Timeslicer v6 (MS6)	PC-DOS	£230
Over-C (MS)	PC-DOS	£225

ADA COMPILERS

IntegrAda	PC-DOS	£645
Janus/Ada Compiler	PC-DOS	£ 95
Janus/Ada Comp. 386-DOS	PC-DOS	£ 95
Janus/Ada Sm'l Mdl comp	PC-DOS	£ 50
Meridian Ada PDK-286	PC-DOS	£935
Meridian Open Ada/386	PC-DOS	£495
Meridian Open Ada	PC-DOS	£260

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, AMSTRAD, ATARIST, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt.

For more information call us.

ASSEMBLERS

2500AD 80386 ASM v5	PC-DOS	£230
MS Macro-86 v5.1	OS/2&PC-DOS	£ 76
MS MASM PDS v6	OS/2&PC-DOS	£ 63
PharLap 386 DOS Ext SDK	MS-DOS	£350
PharLap 386 ASM/Linkloc	MS-DOS	£910
Optasm	MS-DOS	£110
Turbo Debugger v2	PC-DOS	£ 89
2500AD Z80 ASM	CP/M-80	£ 70
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

Not all assemblers are supplied with a linker. Check before ordering.

DEBUGGING TOOLS

Multiscope for OS/2	OS/2	£315
Multiscope for DOS	PC-DOS	£140
Multiscope for Windows	WIN	£270
Periscope I/512K	PC-DOS	£490
Periscope II	PC-DOS	£150
Periscope II-X	PC-DOS	£130
Quaid Analyser	PC-DOS	£130
Soft-Ice	386&PC-DOS	£340
Soft-Ice + Magic	CV 386&PC-DOS	£440
Turbo Debugger v2	PC-DOS	£ 89

PROGRAMMING TOOLS

Ada Compilers	**Algol Compilers**
Assemblers & Libs	**AWK**
Basic Compilers	**Basic Interpreters**
Basic Utilities	**Basic Libraries**
C Compilers	**C Utilities**
C Interpreters	**C Libraries**
C++	**Cobol Compilers**
Comms.Libraries	**Cross Assemblers**
Database Libs.	**Debuggers**
Dis-assemblers	**Editors**
Engineers Libs.	**Expert Systems**
Forth	**Fortran Compilers**
Fortran Libraries	**Graphics Libraries**
Icon	**Linkers/Locaters**
Lisp	**Logo**
Modula-2	**Nial Interpreters**
Object Oriented	**OPS 5**
Pascal Compilers	**Pascal Libraries**
Prolog	**Rexx**
Risc	**Screen Libraries**
Smalltalk	**Snobol**

We stock many items for which there is no space in these advertisements.

THE C LANGUAGE

Refer also to the C++ section.

C COMPILERS

Microsoft C v6	OS/2&PC-DOS	£222
Topspeed C for OS/2	OS/2	£335
Watcom C v8 Prof	OS/2&DOS	£325
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	S-DOS	£275
HIGH C v1.61	MS-DOS	£390
HIGH C 386 Global	PL386&MS-DOS	£650
Hi-Tech C	MS-DOS	£165
Hi-Tech C 386 PDS	PC-DOS	£385
Microsoft QuickC v2.5	C-DOS	£ 50
M'soft QuickC/QuickAsm	PC-DOS	£ 83
MIX Power C & lib s'ce	PC-DOS	£ 30
Topspeed C Std	PC-DOS	£110
Topspeed C Prof	PC-DOS	£165
Watcom C v8 Std	MS-DOS	£260
Watcom C/386 Std	PL386&MS-DOS	£585
Watcom C/386 Prof	PL386&MS-DOS	£780
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Hi-Tech C	Z80 + CP/M-80	£105
Mix C	Z80 + CP/M-80	£ 35
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C 68000	ATARI	£140
Mark Williams C v2	ATARI	£110
Prospero C	ATARI	£ 55

C INTERPRETERS

C-terp V3.5	PC-DOS	£165
Instant C/16M	PC-DOS	£650

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Basic PDS 7.1	**£222.00**
Microsoft C PDS 6.0	**£222.00**
Microsoft C/Windows SDK	**£364.00**
Microsoft Cobol 4.0	**£365.00**
Microsoft Fortran 5.0	**£205.00**
Microsoft Macro Assembler 6.0	**£63.00**
Microsoft MS-DOS 5 Upgrade	**£60.00**
Microsoft Pascal 4.0	**£39.00**
Microsoft QuickBASIC 4.5	**£50.00**
Microsoft QuickC 2.51	**£50.00**
Microsoft QuickC & Assembler	**£83.00**
Microsoft QuickPASCAL 1.0	**£50.00**
Microsoft Source Profiler	**£45.00**
Microsoft Visual Basic	**£95.00**
Microsoft Windows 3.0	**£70.00**
Microsoft Windows SDK 3.0	**£265.00**
Borland C++ 2.0	**£206.00**
Borland Corporate Prod Pack	**£330.00**
Turbo C++	**£50.00**
Turbo Debugger & Tools 2.0	**£89.00**
Turbo Pascal 6.0	**£69.00**
Turbo Pascal Professional 6.0	**£137.00**
Turbo Pascal for Windows	**£103.00**
ObjectVision 1.0	**£79.00**
Paradox 3.5	**£340.00**
Paradox Engine 2.0	**£223.00**
Paradox MultiPack	**£405.00**

WITH FULL TECHNICAL SUPPORT

32-Bit FORTRAN compilers

Salford Software has begun shipping Windows versions of its FTN77/386 and FTN77/486 32-bit FORTRAN compilers. Each compilers includes a run-time library with over 200 routines and there is also a 32-bit DOS Extender. A basic interface to the Windows API (EasyWin) will also be available. FTN77/386 costs £765 and FTN77/486 costs £895. Salford Software can be contacted on 061 7455678.

NetWare Lite

Novell has announced a new low-cost peer-to-peer network operating system. NetWare Lite is designed to enable small businesses to share files and printers and is interoperable with the company's server-based NetWare v2.2 and v3.11. DOS-based PCs are connected via COM or LPT ports. The product is priced at £70 per node (software only) and should be generally available at the end of October. Novell is on 0344 860400.

Stealth

Quaterdeck has announced a new version of its extended memory manager (QEMM-386 V6.0) which Quaterdeck is claiming, 'adds up to 211 KB additional memory'. To achieve this QEMM V6.0 features a new technology called Stealth which effectively remaps the PC's ROM, freeing the address space occupied by the ROM, so that it can be used as high memory. QEMM V6.0 costs £79. Quaterdeck may be reached on 0245 496699.

New look to PVCS

Intersolv has released a new version of PVCS, the popular version control system. PVCS V4.0 is capable of operating across many platforms including DOS, OS/2 and UNIX. It now provides interfaces to Micro Focus' COBOL Workbench and Microsoft's Programmers Workbench. There is also a special SQL interface. PVCS V4.0 costs $1000 per developer. Intersolv can be contacted on 0727 812812.

VR for CG

This year's Computer Graphics conference will concentrate on 'Virtual Reality', the fashionable 3D graphics technique which allows the user to interact with an artificial environment. Some of the world's VR gurus will be attending, including Scott Fisher, who was responsible for pioneering work carried out at NASA. As usual, the conference will be complemented by an Computer Animation Festival, at which some of the latest films will be shown. Details from organisers Blenheim Online (081 868 4466).

C++ Goodies from Borland

Borland has released a number of new C++ products for both the DOS and the Windows environments. First there's Borland C++ and Application Frameworks. This bundles together Borland C++ V2.0 (including both the C and the C++ compilers, the Integrated Development Environment, the WhiteWater Rescource Toolkit, Turbo Profiler and Turbo Assembler) with ObjectWindows and Turbo Vision all into one box (weighing 8 Kg - but it does have a handle). ObjectWindows is Borland's long awaited C++ class library for Windows. Previously only available for Turbo Pascal for Windows (TPW), ObjectWindows now provides C++ programmers with classes to manipulate such objects as windows, dialog boxes, menus and icons. Unlike its Pascal cousin, the C++ library is based on multiple inheritance and appears more complicated. Turbo Vision is a C++ text-only user interface class library for DOS. Previously only available with Turbo Pascal, Turbo Vision is now being shipped with Turbo C++.

Borland is now shipping, as a separate package, its own Windows resource editor (as opposed to Whitewater's) called Resource Workshop which provides an integrated environment for interactively designing resources and editing .RC files. It has a bitmap editor, a dialog box editor a menu editor and a text editor. Borland has also devised a number of custom controls, familiar to users of the TPW IDE, which look far better than the standard Windows controls (eg 'three-dimensional chiselled steel radio buttons and check boxes').

Finally there is a video training course that's called 'The World of C++'. Targeted at C programmers who would like to convert to C++, the course consists of 21 lessons and requires an AT&T 2.0-compatible C++ compiler/translator (eg Turbo C++). Prices: Borland C++ & Application Framework costs £399.95 (and includes the complete source code for ObjectWindows and Turbo Vision), Turbo C++ & Turbo Vision is £99.95, Resource Workshop costs £29.95 and the video course is priced at £99.95. For more information contact Borland on 0734 321150.

DR DOS 6.0

In an attempt to stay one step ahead of Microsoft, Digital Research has released version 6.0 of DR DOS. New features include an improved disk cache; better memory management; a 20-task task-switcher; an online user manual; an undelete facility and the ability to cut and paste between DOS applications. For the brave and tight of disk, an optional data-compression facility enables PCs to double the capacity of their hard disk. The SuperStor program and its associated device driver operate on a partition basis to automatically compress data when you write to the drive and decompress it when you read.

DR says it now has over 10% of the worldwide market for DOS. DR DOS 6.0 is priced at £79, with existing users able to upgrade for £24.99. Contact Digital Research on 0635 35304.

New from Watcom

Watcom has released the latest version of its 32-bit C compiler (C8.5/386). It is packaged with DOS/4GW, a royalty free 32-bit DOS Extender from Rational Systems, which provides up to 32 MB of virtual memory and supports the DPMI, VCPI and XMS standards. There are also 32-bit versions of the Windows API and C Library. Watcom has also announced a new FORTRAN 77/386 compiler. There is a introductory price on both compilers of $795 (usual price $995). For more information contact Watcom on 0101 800 2654555.

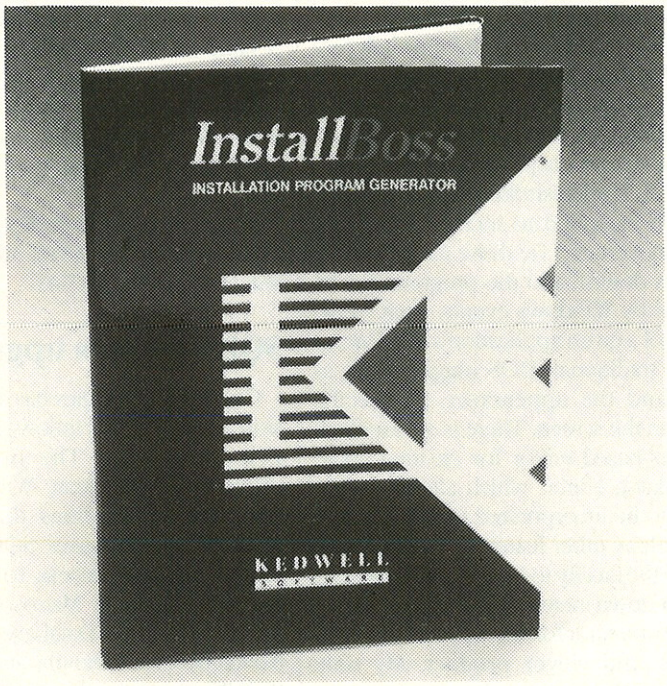
Assembler Library

Quantasm has introduced Quantasm Power Lib V2.0 (QPL) which provides DOS assembly language programmers with over 600 routines, including sub-routines for string handling, sound control and maths functions. QPL requires MASM 5.x, TASM or SLR OPTASM assembers. The library may also be used by C, BASIC, Pascal and FORTRAN programs. Quantasm Power Lib costs $99.95 and $299.95 with source code. For more information contact Quantasm on 0101 408 2446826.

Quintus Prolog Add-ons

AI International Ltd has announced two new products to compliment its Quintus Prolog V3.1 compiler. Quintus Flex is an expert system toolkit which offers developers forward chaining, frame-based hierarchies and a Knowledge Specification Language. Quintus ProWindows V2.0 provides an interface between a PROLOG application and X-Windows, enabling developers to produce GUIs for their Quintus Prolog programs using Open Look 2. Quintus Flex and Quintus ProWindows V2.0 are priced at £1750 each (SPARCstation version - single user licence). For more information contact AI International Ltd on 0923 247707.

Make the best impression with your Boss!



InstallBoss generates installation programs for MS-DOS based computers.

InstallBoss generates professional installation programs and procedures for your software application.

InstallBoss produces programs that are simple and easy to use.

InstallBoss makes the best impression!

InstallBoss is a menu-driven application that creates .EXE installation programs.

The package can be learned in about five minutes and includes sample files to get the user up and running.

As an end-user you require no technical knowledge. You simple type A:Install, and InstallBoss does the rest.

InstallBoss uses self-extracting archive files created with a file compression utility.

The program prepares distribution diskettes for simple, hands-free installation by the end-user.

This means both programmers and end-users save time because less technical support is needed.

InstallBoss:
- no programming
- modifies config.sys
- modifies autoexec.bat
- no script files
- no licence fees
- selective installation
- on-screen, easy help

Telephone: 010-353-1-2804839
Fax: 010-353-1-2805082

Kedwell Software Europe, PO Box 50, Dun Laoghaire Co. Dublin.

CIRCLE NO.177

Turbo Debugging

Paradigm DEBUG/RT from Great Western Instruments is a customised version of Borland's Turbo Debugger that can be used to debug embedded microprocessor systems. There's a range of debuggers providing support for several microprocessors including NEC's V25, V35, V40 and Intel's 80186 and 80188 family. Paradigm DEBUG/RT costs £380 and is distributed by Great Western Instruments Ltd on 0761 52116.

Bjarne Speaks

A new user group, ECUG (European C++ User Group) has been set up to represent the C++ programming community. It will be holding a two-day conference over the weekend of 30th November to 1st December, to address future trends in C++. Bjarne Stroustrup (creator of C++) will be one of several speakers at the conference. For details contact ECUG on 071 2535121.

Hard Disk Protection

It has been hailed 'a Write-tab for your hard disk'. WriteguarD, from Marscott Ecosse Ltd, is a little box of tricks that lives between a hard disk and its disk controller, providing a write protection facility. It enables disk partitions to be protected from being infected by viruses. Switches are used to set the number of cylinders to be protected. It currently supports ST5096, ST412, SCSI and IDE hard disk controllers. WriteguarD costs £120 and is available from Marscot Ltd on 0383 416 089.

Windows Intelligence

Neural Computer Sciences (NCS) has produced the first neural network package for Windows. Neural Desk enables networks to be constructed, trained and then run within the Windows environment. It is even possible to interface Neural Desk with other Windows applications using Dynamic Data Exchange. It also provides a run-time module for embedded applications. Neural Desk costs £985. NCS can be contacted on 0703 667775.

New 486SX speeds

Intel has announced new versions of its 486SX CPUs. In addition to the existing 20 MHz clock speed version, it has now added 16 MHz and 25 MHz units to the range. The Intel 486SX lacks the floating point unit of a fully-fledged 80486 DX, but is identical in other respects. Early 486SX-based machines have been criticised for being poor value when compared to units based on the faster AMD 386.

Automator for Windows

Automator mi, from London-based Direct Technology, is a well established 'software robot' type DOS product. It is based on a TSR program which is capable of loading other applications on top of itself and monitoring and controlling their actions. It can record and play back keystroke sequences, and also incorporates a simple programming language. Automator is typically used to run micro/mainframe link programs; for example it can be programmed to carry out an overnight data download.

Automator for Windows is the freshly-launched equivalent product for the eponymous GUI. The keystroke-recording facility has been extended to include significant mouse movements (ie those made when a button is down), and the program can be triggered by Windows events, such as the resizing of a given application's window, as well as 'traditional' DOS triggers such as timers and the appearance of specific strings on the screen. There is an excellent Windows-based editor for creating code, and a dialog editor, which allows simple dialogs to be incorporated into Automator scripts. Many other features - I particularly admired the facility to insert extra items into the drop-down menus of *other applications* - make Automator for Windows a very well-rounded and clever product. My only qualms are the absence of a screen capturing/comparison system (which would allow the product to be used in automated regression testing of Windows apps) and a slightly over-quirky script language.

Such cleverness doesn't come cheap, though - Automator costs £595 per copy (£495 if you order before the end of October). Contact Direct Technology on 081 847 1666.

Applications Needed

Microsoft's latest COBOL compiler (MS-COBOL V4.5 professional development system) provides three mechanisms for producing COBOL applications under Windows. QuickWin library is the first and is by far the quickest. Text-based COBOL applications only need to be recompiled and relinked in order to reap the benefits of running under Windows. All screen output is handled automatically. In addition, MS-COBOL fully supports Dynamic Link Libraries. Finally there are the tools provided in the SDK which allow complete COBOL applications for Windows to be developed from scratch.

Microsoft COBOL professional development system costs £650. Microsoft can be reached on 0734 500741.

SQLWindows upgrade

Gupta Technologies has announced version 3.0 of its SQLWindows application development tool. The product allows programmers to create Windows 3.0 and OS/2 PM applications that run against multiple SQL databases. SQLWindows currently supports access to DB2, Oracle, Gupta's SQLBase, Microsoft/Sybase SQL Server, OS/2 EE Database Manager and Novell Btrieve. Programs are written in SAL (SQLWindows Application Language) and may be interpreted or saved as .EXEs. Features introduced in V3.0 include enhanced national language support; the inevitable object-oriented extensions and more than 150 new SAL functions.

SQLWindows V3.0 is priced at £1,295. Current users of V2.0 can upgrade for £150. For more information, contact Gupta on 0628 478333.

286 Extensions for TopSpeed

JPI has announced 286 DOS extender technology to support its TopSpeed range of C, C++, Pascal and Modula-2 DOS and OS/2 language products. There are to be two main components. A replacement for the standard TopSpeed environment can run the protected mode OS/2 versions of the language compilers, improving both capacity and compilation speed. The extender toolkit will contain the libraries needed to allow TopSpeed users to generate their own DOS extended applications. These libraries support a subset of the OS/2 kernel API, OS/2-like preëmptive threads, a code and data virtual memory management system and protected mode graphics. The extender supports all three common extended memory management schemes: XMS, VCPI and DPMI.

JPI is accelerating the trend, started by Zortech, towards making DOS Extender technology affordable. For a start, applications built with JPI's extender may be freely distributed without license payment. The company has stated that current owners of DOS compilers, wishing to benefit from the extender technology, may acquire the equivalent OS/2 compilers at cost. Finally, the source code of the extender will also be available, although the price for this has not yet been fixed.

The TopSpeed DOS extender toolkit (provisional name) is due to start shipping on November 1st priced £99, with an upgrade deal for existing TopSpeed users. JPI is on 0234 267500.

Runs on everything... well, almost

Glockenspiel C++ Compilers for UNIX

Glockenspiel C++ runs on the widest range of machines, from 386 PC's to the largest VAX. With over 120 supported machine/operating system configurations, including DEC, RS6000, Suns, SCO and many others, chances are we have the one you want. All Glockenspiel C++ Compilers are AT&T C++ conformant and all Glockenspiel C++ code is consistent across platforms

Glockenspiel CommonView for Motif

CommonView is the standard class library for the rapid creation of Windows and OS/2 PM Graphical User Interfaces in C++. Now CommonView is available for Motif, making it easy to write Motif applications and making those applications portable to the PC.

For full details call The Products Group on 0285 - 655888

glockenspiel C++

CIRCLE NO. 178

QA

Please send more information on:

☐ Glockenspiel C++ for UNIX ☐ CommonView Class Libraries for Motif, Windows or PM
☐ C++ for PC's ☐ Training ☐ Consultancy

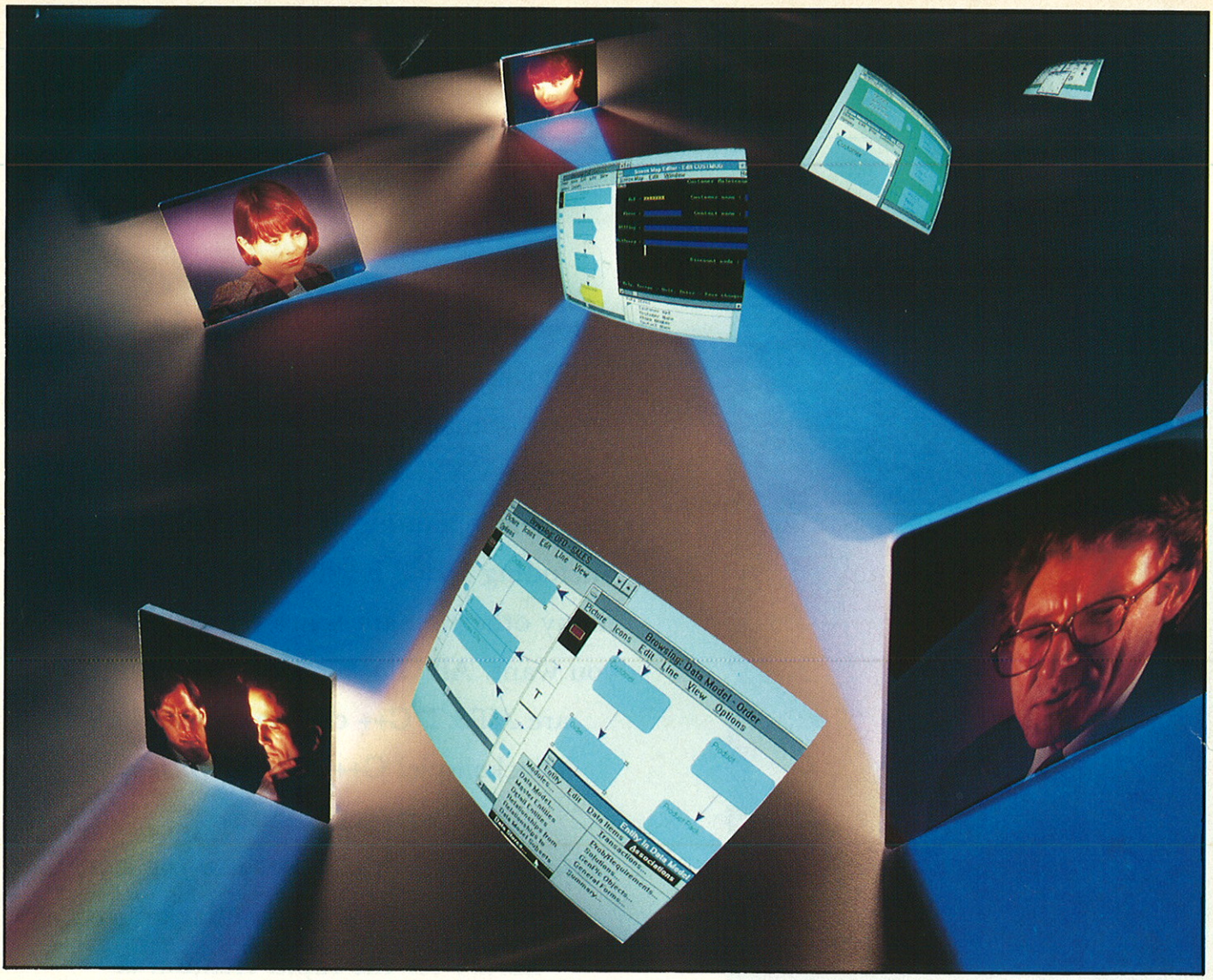
NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

_____ TEL. NO. _____

Mail to: The Products Group, QA Training Ltd., Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.
Tel: 0285 655888 Fax: 0285 640181



LBMS Systems Engineer.

Multi-user CASE to boost your team's productivity.

It's here! Practical CASE automation that takes team productivity to a new high . . . LBMS Systems Engineer. Built on the LBMS pedigree, this multi-user windows-based tool offers leading-edge technology to support the rapid application techniques you want to use.

So why multi-user? Systems Engineer networks your team to provide concurrent and immediate access to shared information. The resulting accuracy and consistency are the keys to rapid progress in application development.

How does Windows help? Windows™ 3 is a breakthrough in performance and usability. Systems Engineer fully exploits this natural desktop approach. As a result, multiple techniques can be used side-by-side to achieve the ideal decision-making context for quality designs. Systems Engineer transforms the PC into a powerful developer's workstation by integrating CASE with your choice of Windows™ tools (word processing, e-mail, project management etc.). For added flexibility, an OS/2™ version will be ready when you are.

As an IBM AD/Cycle™ vendor and partner to other leading industry suppliers, we offer an integrated set of CASE and methods solutions that can be matched perfectly to your development environment.

So that you gain the very best from our CASE tools and methods, we back you with the level of training and support that's appropriate to you. Our aim is to transfer the skills and experience of the best people in the business to your organisation.

Get LBMS integrated CASE solutions working with your team. Phone Elizabeth Baxter now on 071 636 4213 for a demonstration, more information or details of our CASE technology seminars.

Evelyn House 62 Oxford Street London W1N 9LF
Tel: 071-636 4213 Fax: 071-636 2708

LBMS

C++ framework for SQL

Dublin-based C++ software house, Glockenspiel, has released a C++ application framework for accessing a variety of SQL and ISAM databases. CommonBase is the database equivalent of the company's CommonView GUI class hierarchy (*.EXE* August '91). The library contains a set of generic C++ classes (such as `DBDatabase`, `DBTable` and `DBCursor`) that allow programmers to write database-independent code. It looks like pretty sexy stuff and should obviate the need for hard coding table and column definitions with specific data types. Another advantage is that applications can be developed initially using an inexpensive ISAM database (such as Coromandel's ObjecTrieve) and then ported to a high-powered SQL server at a later stage, with only minimal coding changes.

CommonBase is currently available for the Microsoft/Sybase SQL Server and Gupta's SQLBase. Versions for Oracle and ObjecTrieve should be available later this month, with support for Coromandel's Integra SQL in the pipeline. Glockenspiel's own C++ compiler is required, although there are plans to support Borland C++ sometime next year. At present, no plans to support Zortech. Prices start at £399. Glockenspiel is on 010 353 1 733166.

AI tools for Windows

Logic Programming Associates (LPA) has introduced three new software tools for Windows Enhanced Mode. LPA 386-PROLOG for Windows is a 32-bit incremental compiler which operates completely within Windows and includes a dBASE III interface, a 64 bit floating point library and fully supports virtual memory.

Prolog++/386 is an object-oriented version which adds multiple inheritance and message passing to the PROLOG engine. An expert system toolkit (flex/386 for Windows) has also been announced. All three provide interfaces for C and DLLs. LPA 386-PROLOG for Windows is priced at £1,500. Prolog++/386 for Windows costs £2,000 and flex/386 for Windows costs £2,500. For more information contact LPA on 081 8712016.

dBASE2C

Sequiter Software has released a tool that converts dBASE III+ code into C. CodeTranslator V1.0 reads in a dBASE program and then creates a C source file that uses the company's CodeBase 4.2 library to access the database files. Once compiled, the converted application should run faster and take up less memory than the original. Heaven knows what the generated source looks like, but it could be worthwhile if you've already got heaps of dBASE code that would benefit from being souped up.

CodeTranslator V1.0 retails for £129 and requires the CodeBase 4.2 C library (priced at £190). Both are available from The Software Construction Company on 0763 244114.

CASE-based Methodology

Computer & Engineering Consultants Ltd (CEC) has developed a new methodology called Foresight, which it claims is the first one to be fully integrated into a CASE tool for the PC. This enables a user to browse through the methodology, utilising diagrams, menus and online help to present the information.

Foresight is presently available under The Information Engineering Workbench and The Application Development Workbench from KnowledgeWare Inc. A single user licence is available at £12,500. For more information contact CEC on 0276 51414.

PCTV

Word for Windows, Lotus 1-2-3 for Windows and now Eastenders for Windows. Yes, you can now watch telly from the comfort of your own PC. Multimedia vendor, Magnifeye, has released the Screen Machine TV Tuner that allows you to receive a TV signal on a Mac II or PC running Windows 3.0.

The tuner itself is an external device that operates with a Screen Machine digitiser board inside the PC. The TV picture can be viewed on a VGA monitor in a window of any size, and individual frames can be captured. All TV standards are accepted (PAL, NTSC and SECAM) and the frequencies can be adjusted digitally within the software (reception can be via cable or aerial). There's also a C-language API available. Sounds like the ultimate desktop accessory.

The TV tuner itself costs £250, and can be used as a standalone product to display and capture teletext data. It is shipped complete with all cables and software needed to control the device from DOS, Windows or Mac. If you actually want to display a TV picture, you also need a Screen Machine board costing £1,123 for a PC/AT or £1,498 for a Mac II. Magnifeye can be reached on 071 2218024.

AT&T C++ library

After two years of extensive testing, Unix System laboratories (a subsidiary of AT&T) has finally announced the second release of its C++ Standard Components source code library, providing low-level reusable routines for handling such tasks as the manipulation of strings and lists. The C++ Standard Components Release 2 library costs $1500 for a single CPU license and is distributed in the UK by Unix System Laboratories Europe, on 081 5677711.

Actor with SQL objects

Whitewater has announced Actor V4.0 which now provides developers with an object-oriented interface to SQL databases. It also includes a number of DLLs to access Paradox, dBASE and Excel file formats. Actor also offers a safe mechanism for multiple inheritance. Actor V4.0 Professional costs £375 and is distributed in the UK by Neow on 06286 68334.

VB shareware

EMS has begun shipping its VBASIC library of public domain utilities for Microsoft's Visual Basic. For $59.50 you get 11 diskettes with over 100 VB programs. The company also supplies many other shareware libraries, including C++, Windows and Turbo Pascal. If you're interested, EMS can be contacted by telephone on 0101 301 9243594 or fax on 0101 301 9632708.

Network Monitor

Axial Systems has released a low-cost network monitoring system called Comtest NM. The product runs on a PC/XT/AT or compatible fitted with any industry-standard Ethernet network adapter card. Graphical and numerical displays are used to show network utilisation, traffic rates and errors, either for the whole network or for individual nodes. Protocols supported are TCP/IP, ISO, AppleTalk, Novell and XNS. Comtest NM costs £995. Axial Systems can be contacted on 0932 866466.

18 carat techie

Disk Technician Gold is a new utility program that continuously checks and repairs hard disk errors. It runs as a 12 KB device driver (not a TSR) and monitors all disk accesses, looking out for problems. Whenever a critical error (one that would cause loss of data) occurs, it beeps and tells you what to do. Disk Technician Gold runs under DOS and supports ST506/412, SCSI, ESDI and IDE drives. The package costs £124.99 and is available from World Computer Diagnostics, tel 071 5377300.

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.

Poxy Riposte

Sir,

I am not sure what a 'proper' word is, but *poxy* would certainly seem to be one, pace Roger Lee *(see Letters, .EXE September '91).*

It is listed in the supplement to the OED and not qualified as slang or colloquial. Its first recorded use dates from 1922, in *Ulysses.*

(Incidentally, Lee's letter, including its misspelling, is a good example of how people who write letters to editors about language do not know what they are talking about. That does not include me, of course.)

Charles Rowe
London

Hidden cost

Sir,

Recently it has become apparent with the massive volume sales of Windows 3, that Microsoft have done a superb marketing job on their software. This fact amazes me, as the package is just a noddy's guide to DOS, with an attractive user interface (a bit like a Skoda with a Lamborghini body kit).

Urged on by seven figure sales of the software, the selling department at .EXE seem now to have come up with an ingenious way of catching up Microsoft's impressive tally. With more stealth than the F19, the price tag on the top right hand corner of the .EXE magazine front has been covered up more and more in recent times, until now it is never seen.

Could this be because something Mega is regularly happening in the computer industry, or is it with the price gone, nobody knows how much they are paying for the magazine. Surely you yourselves don't think the price of your magazine is too expensive as well as the rest of us?

David Stokes
Prestim Computer Systems Ltd
Wiltshire

PS The above comment is an attempt to win a free .EXE tee-shirt, and is not a genuine criticism of your superb (except Jules May) and fairly priced magazine.

Too little insincere grovelling too late, I'm afraid. The cover price did *reappear in September, and this month has been moved to a new, permanent visible spot (anything to oblige). Although showing a cover price on a subscription-only magazine is in itself a dubious conceit, so I had no hackles about covering it up. Ed.*

Amnesty Success

Sir,

We'd like to thank those readers who made donations to Amnesty International in response to our offer of a two-year on-disk index to .EXE. Thanks too, .EXE for publicising the offer. With the help of other PC magazines, we have raised £250 for Amnesty and the cheques are still rolling in.

Michael S Harper
Codebigh Limited
Hampshire

Quarterdeck again

Sir,

Sorry that I omitted the 12 significant disadvantages of MS-DOS 5 compared to Quarterdeck's QEMM *(Letters, .EXE August 1991),* but I do get bored with doing Quarterdeck's PR work for them for nothing. Anyway, here they are:

1. QEMM occupies between 7 and 9KB less conventional memory than do the combination of HIMEM.SYS and EMM386.EXE

2. QEMM can load DOS file and fcb data structures into UMBs (thereby saving using up to a further 15 KB)

3. QEMM's analysis procedure will identify for you the areas of potential high memory occupied by unused ROMs (usually between 32 and 64 KB, otherwise unavailable to you).

4. QEMM will automatically modify your AUTOEXEC.BAT and CONFIG.SYS to load device drivers and TSRs high, making life a great deal easier for the user.

(Mr Ward goes on to list 13 in all, space prevents us from printing them - Ed.)

So surely, while the introduction of some crude memory management features to MS-DOS 5 will serve to educate whole new areas of the PC market-place in the potential benefits of these techniques, the serious drawbacks to trying to do anything sensible using the Microsoft tools can only serve to even further popularise the more sophisticated products such as QEMM. And it will still be true, as it is now, that every dealer who sells a 386-family PC without QEMM is missing an opportunity, and every customer who buys a 386-family PC without QEMM is missing out - on at least 13 counts.

Andrew R Ward
Award Software Limited
Harpenden

Why is it that Quarterdeck's products have such a strong polarising effect on the programming community? On the one hand there are die-hard fans like Mr Ward, on the other there is the viewpoint encapsulated by the following recently overheard exchange:

Prog1: What do you think will be the most important benefit of OS/2 2.0?

Prog2: We won't have to use effing DESQView.

Any letters throwing light on this phenomenon most welcome - Ed.

The Fortran Case

Sir,

You did an excellent job in setting my article on Fortran 90, but you applied the global edit Fortran => FORTRAN. Now I know that this is your house style and is correct for FORTRAN 77, but the Committee made a definite decision that the new language be called Fortran 90. Is there any chance of your changing your house style?

John Reid
Atlas Centre
Rutherford Appleton Laboratory

Apologies for my ill-informed editing. Our Style Guide will be amended as suggested - Ed

Letters submitted to this page may be edited. The writer of the best letter of the month, as judged by the Editor, will be rewarded by a T-shirt or similar-valued .EXE trinket. The best letter is the one printed first.

UNIPALM TRAINING SERVICES

TCP/IP

Now an accepted standard, TCP/IP is the language of UNIX, the standard operating system now supplied by all major computer vendors. Our lab-style course gives you plenty of hands-on experience, using products such as FTP's PC/TCP, building up a fast, clear understanding of TCP/IP.

NFS

Another open systems connectivity standard, now used and accepted industry-wide. We can teach you, using products such as Sun Microsystems' PC-NFS, to understand the concepts and use and administer NFS systems effectively.

X WINDOW SYSTEM OVERVIEW

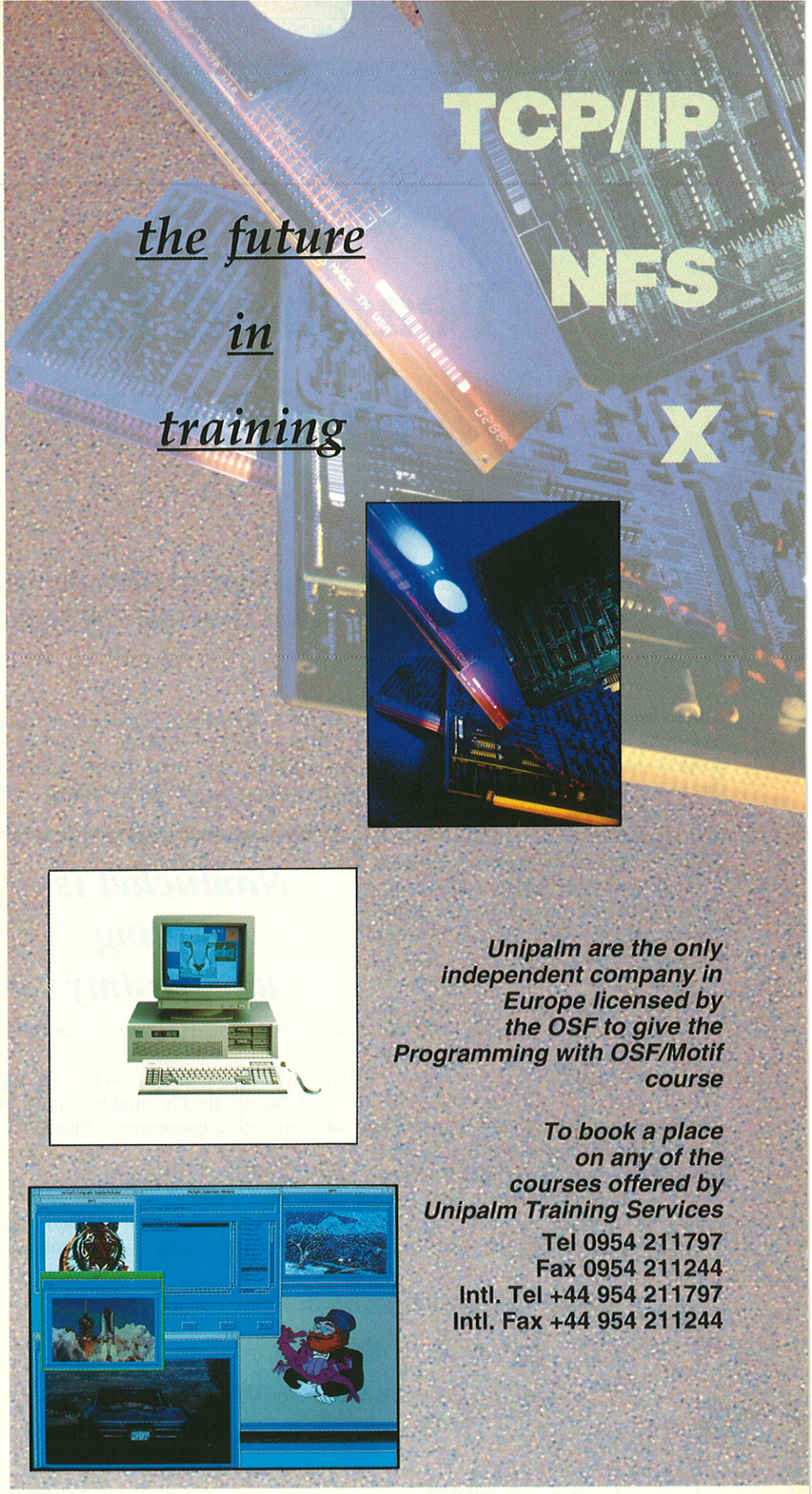
All major software vendors are thinking X now - it's playing a major part in the development of distributed applications. We used our many years of X development, support and marketing experience to develop an X Overview which gives a quick, easy understanding of the principles of X, so you can exploit its power too.

X ON PCS

To provide cost-effective additional X machines, many people use PC-based X servers to convert existing PCs to X terminals. Our PC-based X training includes plenty of hands-on experience – as user and administrator – using products such as PC-Xview from Spectragraphics, and IIM's Windows 3-compatible X server, X11/AT.

OSF/MOTIF AND OPEN LOOK

For people developing X applications, or simply wanting an in-depth understanding of the programming principles behind X, we are the only company in Europe licensed to offer the week-long OSF/Motif programming course. We can also help Sun Open Look X users to get started.



Unipalm are the only independent company in Europe licensed by the OSF to give the Programming with OSF/Motif course

To book a place on any of the courses offered by Unipalm Training Services

Tel 0954 211797
Fax 0954 211244
Intl. Tel +44 954 211797
Intl. Fax +44 954 211244

Unipalm Limited 145-147 St Neots Road Hardwick Cambridge CB3 7QJ England
Tel 0954 211797 Fax 0954 211244 Intl Tel +44 954 211797 Intl Fax +44 954 211244

Unipalm

CIRCLE NO. 180

Fear and Loathing in dBASEland

Borland and Nantucket have unveiled their competing strategies for the future of database development. Paul Kemp and Cliff Saran examine what's on offer.

dBASE and Clipper developers are approaching a watershed. The dBASE language is showing its age and it is no secret that in order to take full advantage of new operating systems, GUIs and databases changes will have to be made. The language is interpreted and, in its raw state, does not easily lend itself to compilation - it simply wasn't designed that way. Sometime around the middle of next year, forward-looking developers will be given the choice of siding with the new guardian of a revamped dBASE (Borland's dBase for Windows), or taking the plunge with Nantucket's Future Technology (NFT) product, codenamed 'Aspen'.

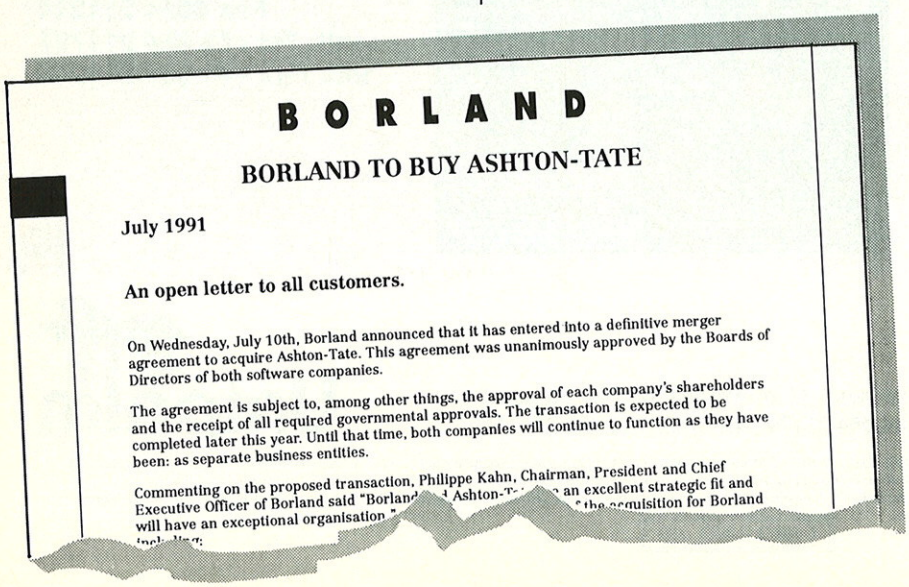
On the surface, both companies appear to be proposing similar features - GUI extensions, OOP and database independence. However, there are significant differences in the detail of what's on offer. Nantucket has thrown down the gauntlet with an aggressive advertising campaign, aimed at undermining the credibility of the newly merged Borland empire. So far, Borland is turning the other cheek and refusing to join the fray, claiming that Nantucket's strategy is stirring up already muddy waters. So what on earth is happening to dBASE?

Nantucket is creating uncertainty

The Borland Vision

With Borland's annexation of Ashton-Tate nearing official completion, the company has responded to a jittery dBASE market by releasing a white paper, outlining its future database strategy. It all looks very impressive indeed. The company goes to great lengths to allay the fears of existing dBASE users by stressing its commitment 'to evolving both Paradox and dBASE for DOS'. More specifically, 'there will be enhancements to dBASE IV 1.1, and new versions, that Ashton-Tate has undertaken to provide'. This means that versions for a variety of UNIX and VMS platforms are still planned. In addition, the dBASE compiler for DOS, that has been under development at A-T for some time now, looks as if it will see the light of day when Borland techies have given it the once-over.

So what is the future of the dBASE language? dBase for Windows (aka Object dBase, formerly Turbo XBase) is Borland's next-generation dBASE compiler and interpreter. It will be a Windows-hosted development environment for the creation of Windows and DOS text-mode executables, and will be fully interoperable with the company's sister product, Paradox for Windows. The two applications will share a common graphical IDE (Integrated Development Environment) and will be able to access each other's file formats transparently. Database independence will be achieved through Borland's 'object-layer', which is a set of C++ classes that present a single interface to the underlying file drivers. Pascal and C++ programmers will be able to write directly to this interface and also use it to create new drivers. SQL servers that Borland is looking to support 'out of the box' include Oracle, IBM (OS/2 Extended Edition and DB2), Sybase SQL Server, Unix Sybase, DEC Rdb and A-T's Interbase server.



BORLAND

BORLAND TO BUY ASHTON-TATE

July 1991

An open letter to all customers.

On Wednesday, July 10th, Borland announced that it has entered into a definitive merger agreement to acquire Ashton-Tate. This agreement was unanimously approved by the Boards of Directors of both software companies.

The agreement is subject to, among other things, the approval of each company's shareholders and the receipt of all required governmental approvals. The transaction is expected to be completed later this year. Until that time, both companies will continue to function as they have been: as separate business entities.

Commenting on the proposed transaction, Philippe Kahn, Chairman, President and Chief Executive Officer of Borland said "Borland ... Ashton-T... an excellent strategic fit and will have an exceptional organisation ... the acquisition for Borland

I asked Rob Dickerson, General Manager of the Database Business Unit for Borland US, about the future shape of dBASE. He replied that the company 'has identified three key enhancements to the dBASE language that will be incorporated into dBase for Windows'. These are: Windows extensions; object-oriented extensions; and the ability to embed C code. The Windows extensions will take the form of standard GUI objects, such as buttons and menus, which will enable the language to operate in a graphical environment. OOP features will be the inclusion of scope (`::`) and member access or 'dot' (`.`) operators, as in C++. When pressed on the subject of inheritance, Mr Dickerson was rather vague, saying that it was 'not possible as such, although you can use the concept of *containership*'. In other words, you can't do it properly - containership merely implies that complex objects can be built from smaller, simpler ones. The ability to embed C code in a dBASE program is designed to overcome the limitations of the language's non-declarative nature when it comes to compiling. Because program variables cannot be given a type in dBASE, a great deal of overhead is incurred at execution-time in determining a variable's data type (this is known as late or run-time binding). The unholy marriage of disparate languages was justified by Mr Dickerson as 'a way of declaring typed variables. We figured that most dBASE programmers already knew C, so it seemed the best choice'. When asked for his reaction to the Nantucket ad campaign, Mr Dickerson confessed to being 'generally amused' but went on 'I think Nantucket is creating uncertainty in its own marketplace with this strategy'.

In not wishing to frighten off the vast installed base of dBASE users, it seems that Borland is shying away from a complete rebirth of the language. By failing to bite the bullet, the company may end up with a C/C++/dBASE hybrid that is unappealing to both the novice dBASE user, and to serious developers who have been constrained by a dated language that was never meant to be compiled in the first place.

The Nantucket Drive

Are you still waiting for a dBASE compiler from Ashton-Tate? At Nantucket they're chanting, 'we have a product that you can compile with *now* and use *now*'. Nantucket has taken full advantage of the current confusion in the market with its cheeky advertisements and has revealed a new strategy which effectively splits the Clipper language product into two branches. Larry Heimendinger, President and COO of Nantucket, told us that 'in order to take full advantage of new technologies, some compatibility will have to be sacrificed'. The result of this is that the Clipper 5.0 architecture will continue to be developed for DOS and the next release will be a fully OOP environment with support for typed variables. Future versions will be fully compatible and add support for event-driven text-mode windowing and DLLs. A character-based version for OS/2 is also planned. The alternate path is a new architecture given the umbrella term Nantucket Future Technology (NFT), with Aspen as the codename for its initial product line. 'NFT takes the language into a whole new area... dBASE won't be a viable option' said the MD of Nantucket UK, George Fletcher. The product will be a superset of the Clipper language and will share only 60-70% code-compatibility with it. It will boast a highly optimised compiler, designed from scratch, and will support multiple GUI platforms (Windows, OS/2 PM, Mac and PenPoint OS are planned), single-inheritance OOP and database independence.

We're not interested in playing 'whose objects are bigger'

When asked about Borland's dBase for Windows, Mr Heimendinger grinned and responded 'We're not interested in playing "whose objects are bigger"... for us, object-orientation is just the starting point - we have it in Clipper 5.0 already'. Fighting talk indeed. Mr Heimendinger seems to have big plans for Aspen, stating that the product is 'aimed at replacing C and C++ for database application development'.

Database independence will be achieved through an object-based approach to data in the language, supported by Replaceable Database Drivers (RDDs) to allow access to different file formats, including a variety of SQL servers. Interestingly, Nantucket itself will not be supplying a Paradox driver because of an alleged licensing tiff with Borland. However, third-party vendors shouldn't encounter this problem and the Babelfish Paradox data driver is already available. Third-party class libraries (like SuperClass) are also available for Clipper 5.0 now, and the number is expected to grow. It would seem that Nantucket has a head-start in the OOP game, with both the Clipper 5.0 architecture and Aspen. In addition, the company claims to have built a new-technology compiler that produces code which runs '10 to 100 times faster than existing Clipper applications'. Borland is not making such extravagant claims for its Windows product, and the company seems curiously unruffled by the challenge. Perhaps it is calling Nantucket's bluff and waiting to see whether Aspen turns out to be anything more than hot air.

The lowdown

It appears that Nantucket will be taking a brave step with Aspen. The company has grown ambitious and no longer wishes its future Clipper and Aspen products to be seen as mere dBASE compilers, but as completely independent language products that also happen to compile dBASE code. If the rhetoric turns into real products, then Nantucket's Aspen would seem to be providing a very rich language for the development of GUI-hosted database applications. However, when it comes to delivering the goods, Nantucket has an unenviable track record - after all, Clipper 5.0 was released well behind schedule. Borland on the other hand has excellent credentials when it comes to OOP and GUIs, and the company is becoming renowned for producing new software at an astounding rate. Ironically though, dBase for Windows does not seem to be pitched at the serious database developer. The prospect of embedding C code in a semi-OOP dBASE language is frankly alarming. No doubt many existing dBASE users will be won over to Borland's dBase for Windows, but if Nantucket gets its act together it could be quids-in with the more discerning developers.

EXE

The trouble with getchar( )

UNIX is an operating system which handles complexities well. But problems can arise when you want to do something simple, as Ben Thompson found out.

Many of us who came to C via other computer languages will remember the thrill of removing the film from our first shrink wrapped compiler and watching with bated breath as it processed `hello.c`, you know, the one that prints 'Hello World'. Surprised that it took up 48 KB on the disk, yes, but generally satisfied with its performance when run. However, when I tried compiling and running my own little piece of code, `myprog.c` (Figure 1) I suffered something of a trauma. The program operated as expected until I hit the enter key. Then all the characters I have already entered spilled out onto the screen again. Where did I go wrong?

I eventually discovered that `getchar()` is in fact a macro which works on the file pointer `stdin` (standard input). This file is buffered, and data read into standard input is not returned to the user until a complete line has been read. The reason why the program seems to be behaving correctly in the first place is that the characters are *echoed* to the screen as you type them. Your `putchar()` calls are not executed until you press enter.

This behaviour is easy to cope with in MS-DOS, generally by using a `bdos()` or `bios` call, but UNIX systems are more complex. Specifically, it is not possible to interrogate the keyboard at a low level, because the keyboard may be attached to a dumb terminal at the other end of a fifty foot wire. All we have to work on is the basic input stream, which is represented by the file pointer `stdin` or the file descriptor `0`. If we want to read characters one at a time as typed, we have to put the input stream into 'raw' mode. UNIX prefers the input stream to operate in line input mode because this saves the kernel work.

Processing terminals in raw mode is very CPU intensive. However, modern users expect modern software. Try selling an office manager something that looks like EDLIN and see how far you get. So any software developer thinking of developing UNIX software, or porting an MS-DOS application in this day and age will almost certainly be working in raw mode. You should be aware that a computer which claims to handle 32 users will probably only handle 16 (or even fewer) when all the terminals are operating in raw mode, but as CPUs get more powerful - and unemployment rockets - this seems to be less of a problem as each day passes.

Going raw

You can put a terminal into raw mode from the shell prompt by typing `stty raw`. You will find that the characters you type are no longer echoed back to you, and that the carriage return key no longer functions. To terminate a line, you need to use a newline, Ctrl-J (next month: how to get the terminal out again).

From the point of view of the C programmer, UNIX provides us with the means to control the behaviour of file descriptors via the `ioctl()` system call. This is a catch-all type of function, which deals not only with flags on input and output streams, but also with issues such as bps rates on serial lines and machine-dependent items like formatting floppy disks. The behaviour of the input stream can be controlled by making `ioctl()` calls on file descriptor `0`.

If you refer to the `setraw()` function in Figure 2 you will see how this can be done. The tilde ~ operator is the C bitwise NOT operator and is used to turn the specified flags off. On input, the flags which map CR to NL are turned off, along with flags which map upper to lower case and strip the high bit on input characters. On output, echo is turned off, along with interrupt detection. For detailed information about this function, and the meaning of the various flags, the best reference is *Advanced Unix Programming* by Marc J Rochkind (Prentice Hall). The original flags are saved and used to reset the input stream to its previous mode when the program exits. Setting the terminal into raw mode in this way has another beneficial side effect, which is that it disables the various hard and soft interrupt keys which will otherwise break into your program (like Ctrl-C in MS-DOS).

termcap

Putting the terminal into raw mode allows your software to read each character as it is typed on the keyboard. This is fine for simple alphanumeric keys, upper and lowercase characters numbers and punctuation etc. However, the situation is more complicated when terminals have function keys and arrow keys, since different terminals may return different codes or character strings for the same keys. UNIX traditionally handles this via a text file database called `termcap`, held in the `/etc` directory. A lot of people find the function of this file confusing. It does not control the terminals on a UNIX system, all it does is tell any software which may be interested (such as `vi`) what codes can be used to control a particular terminal, and what character strings will be returned by that terminal if a function key is pressed.

```
#include <stdio.h>

main()
{
  int c;

  while( (c=getchar()) != EOF )
    putchar(c);
}
```

Figure 1 - myprog.c

If you could create your own computer for developing software what would you build?

You'd need a system to increase your productivity, one that helped you create applications in half the time it takes you now – and complete projects ahead of schedule!

You'd want a true multitasking system with virtual memory. Because that would give you the kind of computer that will let you compile a program in one window while debugging in another, so if your new application should crash your whole system won't die.

THE ULTIMATE DEVELOPMENT MACHINE.

You'd give it plenty of power and speed to handle even the most complex projects, complete with UNIX and the Motorola 68040 processor.

With your users in mind, you'd want an Interface Builder so you could create an outstanding user interface – quickly and easily.

And you'd want an object oriented programming environment so that you could reuse your software components and it would help if it came with a complete object library.

Fortunately, there's a computer that already offers this, the NeXT computer. The tools provided by the NeXTstep operating environment make it easy to develop advanced, user-oriented programs.

So take the NeXTstep and call 081-565 0005 for more information, or fill out the coupon.

The Professional's Workstation

CIRCLE NO. 181

I'd like to receive more information on NeXT ☐

I'd like to receive more information on NeXT Developer Courses ☐

Please complete and return to: NeXT, 286 Bath Road, West Drayton, Middlesex UB7 0DQ.

Name: _____

Company: _____

Address: _____

_____

County: _____

Telephone: _____ Post code: _____ EXE 10/91

If you look at the entry for `termcap` in the UNIX manuals, you will see that there are a lot of terminal capabilities defined, several pages in fact. This is partly due to the power and flexibility of UNIX, but mainly because UNIX is a bone which has been gnawed by many dogs. A lot of the capabilities represented are very obscure - some refer to terminals last seen as props in Flash Gordon films. Some are rare and fancy, such as the ability to scroll different regions of the screen independently. I once saw a beta test version of a famous UNIX word processor do a panic termination because the terminal I was using didn't have a character delete capability.

The point is that the use of all termcap capabilities is optional, and the fewer of them you use, the more terminals your software will run on. If you are writing an editor, it may be worth checking whether there is a line delete or line insert capability, because these operations will execute much faster if carried out in the terminal hardware. But it is prudent to code the software in such a way that it can redraw the screen if these capabilities are not present. Some capabilities do have to be assumed. Complex software can't run without an absolute cursor addressing capability, for example. If your software is going to be shrink wrapped and sold in Tesco's you may want to do a panic termination if the cursor motion string can't be found, otherwise these problems are normally solved during installation.

Figure 2 shows a complete basic terminal driver module which works with XENIX and UNIX systems to allow raw input, function key detection and access to basic terminal capabilities, clearing the screen, clearing a line, setting the cursor and putting the screen into and out of reverse mode. These are all the capabilities necessary for writing, for instance, quite an adequate word processor. The module has been written with a restricted number of entry points to emphasise the fact that although the methods involved in doing simple I/O operations in UNIX may seem unnecessarily baroque to someone coming from MS-DOS, once solved they can be parcelled off and forgotten about and life goes on much as before.

init_term( )

The terminal is initialised via the function `init_term`. This does two things. It sets the terminal into raw mode via `setraw` and reads the terminal information database `/etc/termcap` for the codes needed to drive the terminal. The entry into the database is determined from the envi

```
/* uio.c : Basic raw mode terminal I/O
           for XENIX and UNIX
   Author: Ben Thompson

This module covers the basics needed to drive
a terminal on a UNIX or XENIX system. It
covers raw mode keyboard input with function
key translation using a method which allows
the ESCAPE key to be detected. It also covers
basic terminal capabilities such as clearing
the screen, absolute cursor motion and
inverse video. More complex functions are
simple to add.

ENTRY POINTS:
init_term()     Set terminal into raw mode and
                interrogate termcap
clear_term()    Restore previous terminal mode
                prior to exit
fetchchar()     Fetch one character with
                function key translation
clear_screen()  Clear terminal screen
clear_line()    Clear to end of line
standout()      Put terminal into
                standout mode
standend()      Put terminal out of
                standout mode
setcursor()     Set absolute cursor position
*/


#include <stdio.h>
#include <fcntl.h>
#include <termio.h>
#include "uio.h"

/* wait 1/20 second between reads
                    in non-blocking mode*/
#define NAPINTERVAL 50
/* try five times, altogether 1/4 second */
#define MAXNAPS 5


/* function definitions */
char *calloc(),*getenv(),
     *tgetstr(),*tgoto();
unsigned int isfk();


LOCAL int charwait; /* holds character
                              pending input */
LOCAL int raw;      /* set TRUE in raw mode */
                    /* input buffer queue */
LOCAL char queue[QUEUE_LENGTH];
   /* Buffer for saving flags on std input */
LOCAL struct termio flagsave;
                /* terminal info structure */
LOCAL struct scrdata sc;


GLOBAL init_term()
{
  setraw();
  getterm();
}
GLOBAL clear_term()
{
  trestore();
}
GLOBAL fetchchar()
{
  int c,m;

  if( queue[0] == EOS )
  {
    for(;;)
    {
      c = inchar();
      putqueue(c) ;
      switch(m = isfk())
      {
        case MAYBEFK :
      /*-- keep checking, is fn key? --*/
          continue ;
        case NOTFK :
      /*-- leave last char, throw out rest */
          break ;

        default :
      /*-- yes, it is a fn key! -- */
          queue[0] = m ;
          queue[1] = EOS ;
          break ;
      }
      break ;
    }
  }

  c = (queue[0] & 0xff);
  strcpy( queue, queue+1 );
  return c ;
}
GLOBAL clear_screen()
{
  fprintf(stdout,"%s",sc.cls);
  fflush(stdout);
}
GLOBAL clear_line()
{
  fprintf(stdout,"%s",sc.eol);
  fflush(stdout);
}

GLOBAL standout()
{
  fprintf(stdout,"%s",sc.sr);
  fflush(stdout);
}
GLOBAL standend()
{
  fprintf(stdout,"%s",sc.sro);
  fflush(stdout);
}
GLOBAL setcursor(l,c)
{
  fprintf(stdout,"%s",
          tgoto( sc.scm , c-1 , l-1 ));
  fflush( stdout );
}
LOCAL inchar()
{
  int c ;

  if( charwait )
   {
    c = charwait;
    charwait = 0;
   } else
    c = rchar();

  return c;
}
LOCAL rchar()
{
  int c;

  read(0,&c,1);
  return (c&0xff);
}

static unsigned int kdfs[MAXKDEFS] = {
  FK1, FK2, FK3, FK4, FK5,
  FK6, FK7, FK8, FK9, FK0,
  };

LOCAL unsigned int isfk()
{
  register int  j ;

/* First check and see if the characters
   in the queue are an exact match
   for any of the function key strings */
  for( j = 0 ; j < MAXKDEFS ; j++ )
    if( !(strcmp( sc.fks[j] , queue )))
      return kdfs[j];

/* Exact matches have been filtered out
   above. Partial matches result in
   return of symbolic value MAYBEFK */
  for( j = 0 ; j < MAXKDEFS ; j++ )
    if( !(strncmp( sc.fks[j] , queue ,
                              strlen(queue))))
      goto maybe;

/* Exact matches on the arrow keys */
/* Return Wordstar-like Ctrl-E,S,D,X */
  if( !(strcmp( sc.kd , queue )))
    return 'X'-64 ; /* Down arrow */
  if( !(strcmp( sc.ku , queue )))
    return 'E'-64 ; /* Up arrow */
  if( !(strcmp( sc.kl , queue )))
    return 'S'-64 ; /* Left arrow  */
  if( !(strcmp( sc.kr , queue )))
    return 'D'-64 ; /* Right arrow */

/* Now check for partial matches */
  if( !(strncmp( sc.kd , queue ,
                              strlen(queue))))
    goto maybe ;
  if( !(strncmp( sc.ku , queue ,
                              strlen(queue))))
    goto maybe ;
  if( !(strncmp( sc.kr , queue ,
                              strlen(queue))))
    goto maybe ;
  if( !(strncmp( sc.kl , queue ,
                              strlen(queue))))
    goto maybe ;

/* If not exact or partial match,
   not a function key hit */

  return NOTFK;
```

Figure 2 - uio.c

"What will happen to dBASE?"

"When can I compile my dBASE code?"

"Will I have to learn Paradox?"

"How can I learn OOP at my own pace?"

"What will the next version of dBASE be like?"

"Will I have to learn another language?"

"Where's the professional compiler?"

"X" what?"

"Will there be a next version of dBASE?"

ronment variable TERM which is set at login time from entries in the system configuration files. The correspondence between a user account and a terminal is entirely pragmatic. `init_term()` has to assume that the software is running on the type of terminal represented by the TERM variable, otherwise all bets are off, but this is an issue for the system administrator to resolve.

The `setterm()` function gets the terminal type from the environment, and reads the termcap entry. It then reads a series of strings into the terminal information structure (which is defined in the header file `uio.h` Figure 3). At this point, adding other capabilities is clearly a piece of cake. Reference to the UNIX manuals will give an idea of the capabilities available, but a better way to find out is to actually look at the termcap entries for your terminals and compare them with the codes in the terminal manual. Some terminals, for example, are capable of a dual intensity display, either high or low. I have seen these capabilities represented by the termcap entries 'h7' (high intensity) and 'i7' (normal intensity), though this is not documented in the UNIX manuals.

A point to note is that where a particular capability is missing, the function initialises the field as a zero length-string. This gives a lot of flexibility to the software, in that if the screen reverse capability is missing the messages simply come up in normal video. With careful programming, software can be organised so that it looks good on screens without dual intensity and even better on screens which do have that capacity. Function key strings are read from one to nine and then zero, which is function key 10. Not all termcap entries map function key one to entry one, some map from zero to nine. This can provide headaches for the installation team. The order represented seems to be the most common one.

The `clear_term()` function just restores the terminal to the mode it was in before `init_term()` was called.

fetchchar( )

This function fulfils the two necessary tasks. It fetches characters from the input one at a time, as typed, and it translates function key presses into symbolic values. It doesn't really matter what these values are, as long as they can't actually be typed at the keyboard. The problem with function key sequences is that there is no way of telling when a key value is received whether it is part of a function key string or not. The strategy adopted here is to check each character coming in to decide whether it might be the start of a function key string. If it might be, then further characters are fetched and stored in an input buffer until either a definite match is found, or the characters stored in the buffer fail to match with any of the function key, or special key strings. If the match fails, the characters typed remain in the input buffer and are returned normally by `fetchchar()`. If the string of characters in the buffer exactly matches a function key string, they are removed from the buffer and replaced with the symbolic value assigned to that key.

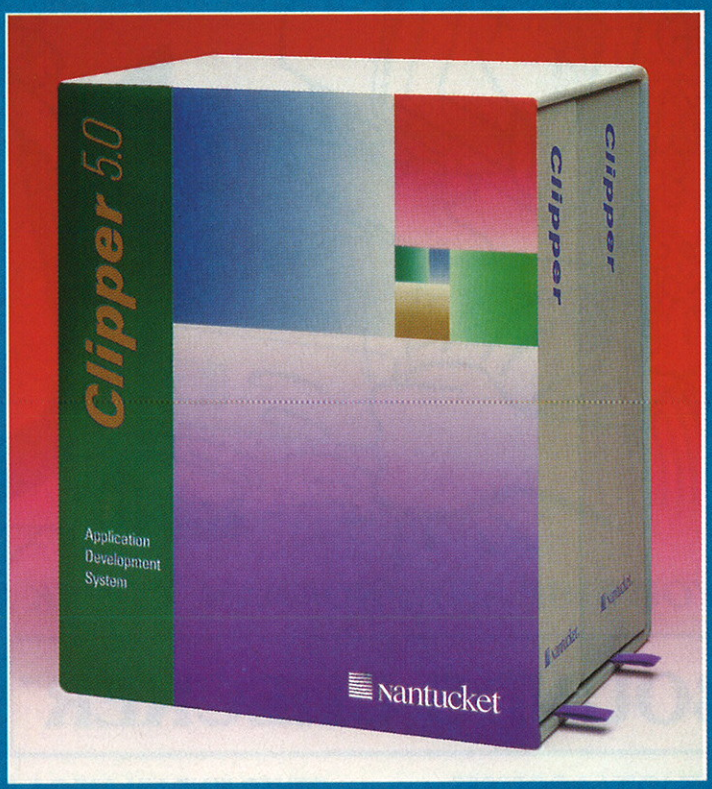
The drawback to this scheme is that if your function key definitions start with 'A', then every time you type an 'A' the process will loop on input waiting for the next character, your 'A' won't be returned to the calling program, and you won't see it appear on the screen until you type at least one more character. To get round this, a non-blocking input function `cready()` is called to find out if another key has been pressed.

`cready` uses the `fcntl()` system call to set file descriptor 0 into non-blocking mode. Normally if you issue a read call on file descriptor 0, the kernel blocks until at least one byte has been entered at the keyboard. However, if file descriptor 0 is set

```
maybe:
/* If a FN key string is coming in, there
   should be another character waiting. If
   not, input is NOT a function key */
  if( !(cready()))
    return NOTFK;
  return MAYBEFK;
}

LOCAL getterm()
{
  int   s;
  char  t[100],*tp,*terminfo;

  if( !(terminfo = calloc(1,1025)))
    dx("sinits");

  /* IF no term variable, try ansi */
  if( !(tp = getenv("TERM")))
    strcpy(t,"ansi");
  else
    strcpy( t , tp );


  s = tgetent( terminfo , t );

  if( s < 1 )
    dx("\n Cannot define  terminal ");

  /* Standout (reverse video) */
  tp = sc.sr;
  if( !tgetstr("so",&tp))
    sc.sr[0]=EOS ;

  /* Standout ends */
  tp = sc.sro;
  if( !tgetstr("se",&tp))
    sc.sro[0]=EOS ;

  /* Clear to end of line */
  tp = sc.eol;
  if( !tgetstr("ce",&tp))
    sc.eol[0]=EOS;

  /* Clear screen */
  tp = sc.cls ;
  if( !tgetstr("cl",&tp))
    sc.cls[0]=EOS;


  /* -- function keys -- */
  tp = sc.fks[0] ;
  if( !tgetstr("k1",&tp))
    sc.fks[0][0]=EOS ;

/** 1 to 8 (omitted) are similar,
  "k2" maps to sc.fks[1],
  "k3" to sc.fks[2] etc **/

  /* "k0" maps to sc.fks[9] */
  tp = sc.fks[9];
  if( !tgetstr("k0",&tp))
    sc.fks[9][0]=EOS ;

  /* -- cursor keys -- */
  tp = sc.ku ;
  if( !tgetstr("ku",&tp))
    sc.ku[0]=EOS ;
  tp = sc.kd ;
  if( !tgetstr("kd",&tp))
    sc.kd[0]=EOS ;
  tp = sc.kr ;
  if( !tgetstr("kr",&tp))
    sc.kr[0]=EOS ;
  tp = sc.kl ;
  if( !tgetstr("kl",&tp))
    sc.kl[0]=EOS ;

  /* --- cursor motion string --- */
  tp = sc.scm ;
  if( !tgetstr("cm",&tp))
    sc.scm[0]=EOS ;

  free(terminfo);

}
LOCAL setraw()
{
  struct termio flagbuff;

  if( (ioctl(0,TCGETA,&flagbuff))==-1)
    dx("\nsetraw ioctl");
  flagsave = flagbuff;
  flagbuff.c_iflag &=
    ~(INLCR | ICRNL | IUCLC |
                ISTRIP | IXON | BRKINT );
  flagbuff.c_oflag &= ~OPOST;
  flagbuff.c_lflag &= ~(ICANON | ISIG |
                                  ECHO );
  /* Min no of characters to wait for */
  flagbuff.c_cc[4] = 1;
  flagbuff.c_cc[5] = 1; /* Time */
  if( (ioctl(0,TCSETAF,&flagbuff))==-1)
    dx("\nsetraw ioctl2");
  raw = TRUE ;
}
LOCAL trestore()
{
  if( (ioctl(0,TCSETAF,&flagsave))==-1)
    dx("ioctl3");
  raw = FALSE ;
}

LOCAL cready()
{
  int flags ;
  int  c , i ,j ;

  /* Save flags in filehandle 0 */
  flags = fcntl(0,F_GETFL);
  /* Set into NDELAY mode */
  fcntl(0,F_SETFL, flags | O_NDELAY);

  for(j=0;j<MAXNAPS;j++)
  {
  i = read( 0 , &c , 1 ) ;
  if( i > 0 )
   {
    charwait = c;
    break;
   }
  nap((long)NAPINTERVAL);
  }
   /* restore flags */
  fcntl(0,F_SETFL, flags & ~O_NDELAY);
  return i; /* 0 if no character waiting */
}
LOCAL putqueue(c)
int c;
{
  int l;

  if( (l=strlen(queue)) >= QUEUE_LENGTH-1 )
    dx("\nQueue overflow");
  queue[l++]=c;
  queue[l]=EOS;
}
/* Panic exit routine */
GLOBAL dx(s)
char *s;
{
  printf(s);
  exit(1);
}
```

Figure 2 - uio.c (continued)

A.



Clipper© 5.0

The Application Development Standard

- Open Architecture Language.
- Pre-defined Objects.
- Compatible with existing Clipper and dBASE III Plus code.
- High-performance network support.
- No, you won't have to learn another language.
- Pioneering virtual memory manager
- Build and run applications larger than available memory.
- Integrates code from C, Assembler dBASE & Pascal.
- Compiles Clipper or dBASE code today.
- Language & Compiler have been standards for years.
- "Nuff said!"

Nantucket

Send for your FREE copy of our booklet "Clipper 5.0 for dBASE Developers"

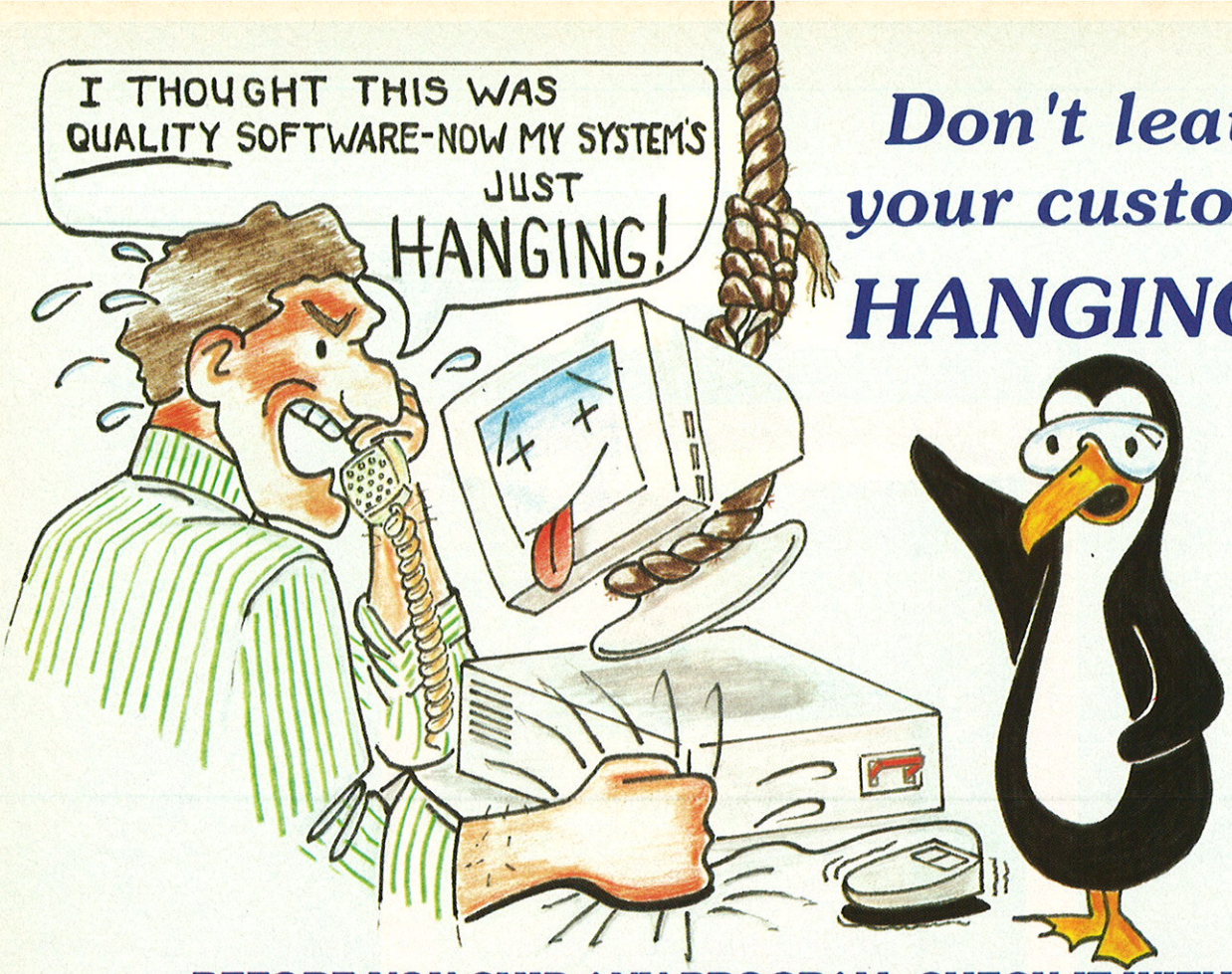
Name: ..... Title: .....

Company: ..... Address: .....

..... Postcode:..... Telephone: .....

Please return to: Nantucket UK Ltd, 8 Bridgegate Centre, Welwyn Garden City, Herts. AL7 1JG.

CIRCLE NO. 183



Don't leave your customers HANGING. . .

BEFORE YOU SHIP ANY PROGRAM, CHECK IT WITH

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1990 COLA

BOUNDS-CHECKER™

WINNER PC MAGAZINE BEST OF 1990

JANUARY 15, 1991 PAGE 168

FINDS OUT-OF-BOUNDS MEMORY ACCESSES *AUTOMATICALLY*

Your program may have 10,000 to a million lines of code. It may occasionally hang mysteriously or it may appear to run flawlessly every time. But under DOS, how can you ever be sure that your program is not corrupting memory it does not own? The only way to be 100% sure is to BOUNDS-CHECK before you ship.

To use BOUNDS-CHECKER you build your program with debugging information (we support most compilers including Microsoft, Borland & JPI). Then you just type <BC file-name>. BOUNDS-CHECKER sets up the 386™ /i486 for protection and lets your program fly. If your program accesses memory it does not own or overwrites its own code, BOUNDS-CHECKER pops up displaying the offending SOURCE-LINE or instruction.

Programming under DOS is a gamble, so why not stack the odds in your favor--CALL TODAY.

(603) 888-2386

Call by 4:00 PM EST TODAY and ask us to EXPRESS you an info packet. In most cases you will receive it by 10:30 AM tomorrow. (USA only)

OTHER FINE NU-MEGA PRODUCTS. . .

The ultimate systems debugger.

Debug:
- Interrupt routines
- Device drivers
- T&SRs
- ROMS
- Applications
- Overlays

Soft-ICE™

Features:
- Break out of a hung program
- Real time Break-Points
- Back-Trace history
- Works with other debuggers

If you are debugging an application, Soft-ICE is seamlessly integrated with BOUNDS-CHECKER so you can easily go back and forth between BOUNDS-CHECKing and debugging: a combination many programming professionals can't live without.

1988 BYTE AWARD OF DISTINCTION

Run CODEVIEW for Windows on a single monitor
CV/1.....$129
• Runs in a window • No annoying flash •
• Runs on any display that supports windows •

™ Nu-Mega TECHNOLOGIES INC
P.O. Box 7780 • Nashua, NH • 03060-7780 U.S.A.
(603) 888-2386 • Fax (603) 888-2465

BOUNDS-CHECKER $249
Soft-ICE $386

Special Offer
Buy BC & S-ICE .....Save $100

30 Day
Money Back Guarantee

CIRCLE NO. 184

```
/* Fundamental Defs and decls */
#define TRUE 1
#define FALSE 0
#define LOCAL static
#define GLOBAL

#define EOS '\0'      /* End of string char
*/
#define ESCAPE 0x1b   /* Esc key return */
/* - */

/* Function Key definitions */
#define FK1 0xb8
#define FK2 0xb9
#define FK3 0xba
#define FK4 0xbb
#define FK5 0xbc
#define FK6 0xbd
#define FK7 0xbe
#define FK8 0xb1
#define FK9 0xb2
#define FK0 0xb6
/* */

#define QUEUE_LENGTH 200
#define MAXKDEFS 10
#define TCAPLEN 10

#define MAYBEFK -2
#define NOTFK -1

/* This structure contains basic data needed
   to drive a terminal. Most terminals
   should support at least these attribs */

  struct scrdata  {
  /* -- reverse video -- */
  char  sr[TCAPLEN];
  /* -- ditto off -- */
  char  sro[TCAPLEN];
  /* -- screen clear sequence -- */
  char  cls[TCAPLEN*2];
  /* -- erase to end of line sequence -- */
  char  eol[TCAPLEN];
  /* -- cursor movement string  -- */
  char  scm[40];
  /* -- function keys as 0 - 9 -- */
  char  fks[10][10];
  /* -- cursor arrow keys -- */
  char  ku[10];
  char  kd[10];
  char  kl[10];
  char  kr[10];
  };
```

Figure 3 - uio.h

into O_NDELAY mode, the read call immediately returns 0 unless a character is actually waiting on input. `cready()` operates with a timeout loop, so that if another key press doesn't emerge after a reasonable length of time the original character can be returned and the software can get on with its business. It is similar to BASIC's `inkey$` function, except that it doesn't return the actual character pressed. A call to `cready()` followed by a call to `inchar()` would be functionally equivalent. In this implementation `cready()` waits for a fixed length of time by using a combination of a loop and a `nap()` system call (`nap()` is better than straight looping, because the kernel can use the time). This allows `cready()` to return quickly if a character is already waiting, but gives a reasonable time for a character to arrive before timing out.

The timeout established in the example is longer than needed for a single user at the console, but delays in transmission of function key strings can be introduced by long serial cables. There is a case for making this timeout dependant on an environment variable, but in practice the value given works well enough. A point to note is that although there are possible input strategies which involve leaving the input stream in non-blocking mode, this makes the program impossible to debug, because if you set a debugger breakpoint you are immediately dumped back at the login prompt.

`fetchchar()` works very well. Fortunately most terminals start their function key sequences with an ESCAPE or control character, so function key processing is not invoked too often. In principle, however, this strategy will cope with any function key codes, including straightforward strings like 'ABCD'. In this case you would experience a slight delay every time you typed an 'A'.

Screen handling

The screen handling functions are extremely simple. Most terminal control is effected by copying the relevant strings to standard output. It is necessary to flush standard output with `fflush(stdout)` for consistent results. The cursor motion function `setcursor()` makes use of a termcap routine `tgoto()`. This does the complex work of translating the cursor motion string copied from termcap into a string of characters which will actually position the cursor. This can be quite a complex operation, since some terminals like the line position first, some like the column position first, some like the co-ordinates in ASCII, others in binary with an arbitrary number added. This is one of those occasions where one is happy to trust AT&T. However, whether you want the cursor addressing to start from 0,0 or 1,1 is clearly something you can determine yourself.

The test program (Figure 4) is a trivial example of how the I/O module can be invoked and used. A point to note is that this I/O module allows an Esc key press to be detected even on an ANSI terminal which uses Esc as the initiator for function key codes. When you compile and run this program, you will note a short delay when you hit Esc, this is the timeout on `cready()`.

```
/** Sample test program to demonstrate
    terminal handling and function
    and escape key handling.

MAKEFILE ENTRY
uio: uio.o test.o
  cc -o uio test.o uio.o -ltermcap -lx
uio.o test.o: uio.h
**/

#include <stdio.h>
#include "uio.h"

main()
{
  int c;

  init_term();
  clear_screen();

  for(;;)
  {
   switch(c = fetchchar())
    {
    case 'X':
    case 'x':
      clear_term();
      exit(0);
    case FK1:
      setcursor(1,10);
      standout();
      printf("You Pressed Function Key 1");
      standend();
      fetchchar();
      setcursor(1,10);
      clear_line();
      break;

    case ESCAPE:
      setcursor(5,10);
      standout();
      printf("You Pressed ESCAPE ");
      standend();
      fetchchar();
      setcursor(5,10);
      clear_line();
      break;
    default:
      putchar(c);
      fflush(stdout);
      break;
    }
  }
  exit(0);
}
```

Figure 4 - Test program using functions in 'uio.c'

Conclusion

For software developers and programmers coming to UNIX or XENIX for the first time from MS-DOS, the problems may seem more formidable than they really are. There are large areas where UNIX and MS-DOS are entirely compatible from a C programmer's point of view. Conversely, there are a few well-defined areas where they are not. Porting software to UNIX will probably involve a balance between writing sections of operating system specific code and rewriting existing code to be less OS specific. Dealing with simple I/O problems is one area of incompatibility that can hold up a project and soak up a lot of time, money and effort, but once these fundamental interfaces are built they can generally be forgotten.

EXE

Ben Thompson is a freelance programmer/consultant and author of the OPTIC opticians recall system developed and marketed by Integral Business Systems. He has been programming since CP/M ruled the market-place and UNIX was what looked after harems. He can be contacted on 0787 269786.

A blank disk and an SAE sent to the Editorial office in an envelope marked 'TERMCAP', in accordance with the instructions on Page 1 Column 1, will secure an MS-DOS disk version of the code given in this article. The rest is up to you.

System V Release 4

Will the real UNIX please stand up!
Simon Kenyon investigates one version of UNIX's claim to that title, System V Release 4.

One definition of UNIX is 'that which runs on Dennis Ritchie's machine'. While this is perhaps a bit whimsical, it does illustrate both the main strength and at the same time the main weakness of UNIX. That is, because of the tremendous flexibility of UNIX, there is not one version. There is an immense variety of them, all clamouring for the user's attention. The problem with this definition of UNIX is that Dennis no longer uses UNIX. He has moved on to OS pastures new in the shape of Plan 9, named after the undisputed world's worst film *Plan 9 from Outer Space.*

Meanwhile, back on Planet Earth, the poor luser (the 'l' is silent) is faced with a decision: which version of UNIX to use? This dilemma is not resolved by choosing a particular piece of hardware, because in this age of commodity software there are multiple OSs available for a given platform.

Take a typical software developer. Chances are that he has a PC of some description. Should he use MS-DOS and Windows 3, or should he broaden his horizons? If you are faced with such a decision, you need to know the facts about UNIX System V Release 4.

History

The UNIX Operating System grew out of the demise of the Multics Project at AT&T Bell Labs in the late 1960s. In fact it was this association with Multics that gave UNIX its name. That the name UNIX implies 'single-user' is ironic, given that UNIX is the OS of choice for users wishing to escape the restrictions of the real single-user OS, MS-DOS.

It was not until 1973 that UNIX saw the light of day outside AT&T. Because of the US Justice Department's 1956 Consent Decree, AT&T was forbidden to get into the software business, so UNIX was licensed to universities for a nominal sum ($400). UNIX had by this stage reached Version 5 (or 5th Edition - meaning the edition of the manual, which was the final arbiter in those days). AT&T announced System III in 1980, and System V in January 1983.

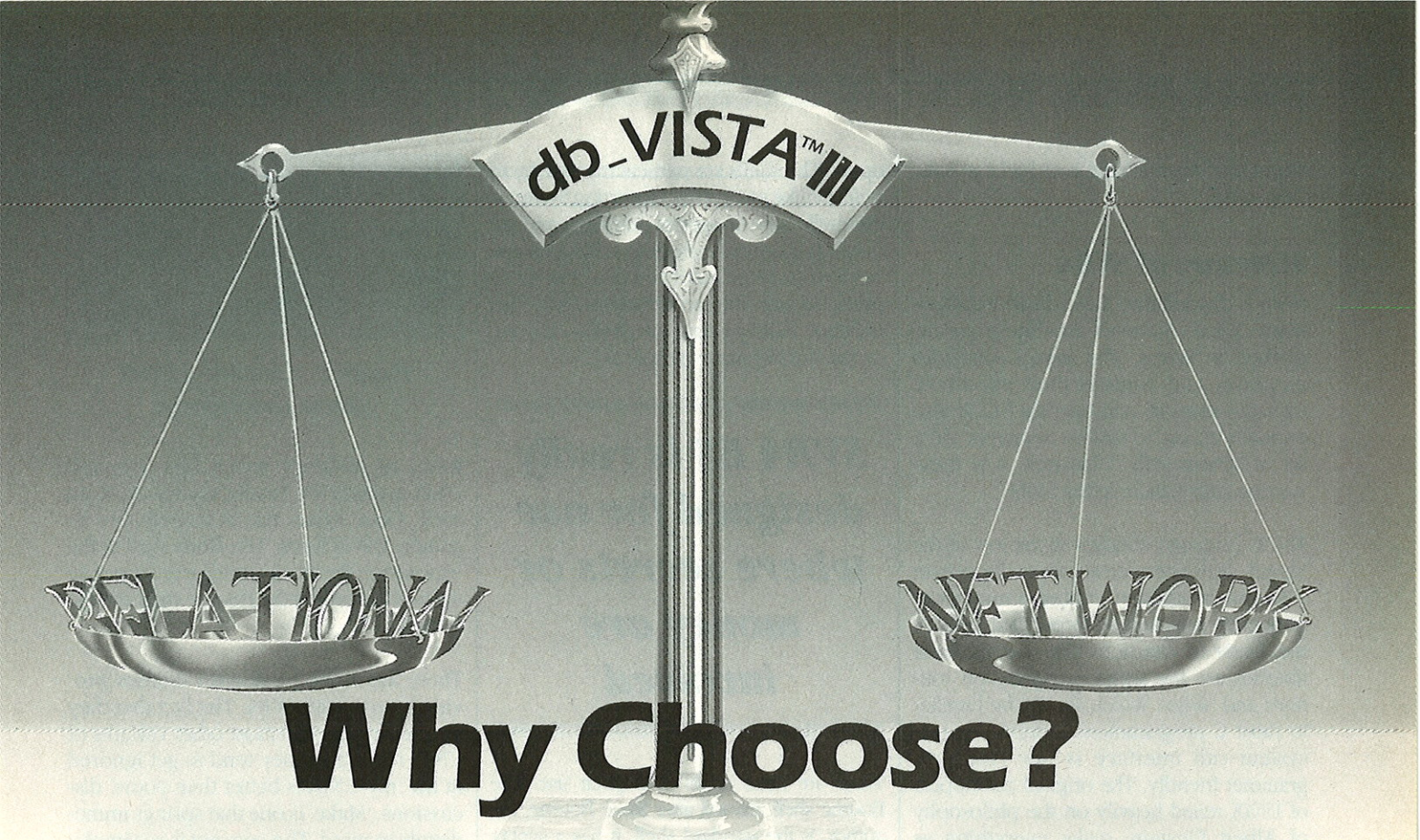
Meanwhile, back on Planet Earth, the poor luser (the 'l' is silent) is faced with a decision

Figure 1 is a UNIX family tree. As can be seen, SVR4 draws together a number of different versions of UNIX; principally System V Release 3, BSD and Xenix.

Who owns UNIX?

At the beginning of April, AT&T sold a portion of UNIX System Laboratories (USL) to the 11 computer companies listed in Figure 2. AT&T has stated that this selling-off process will continue until such time as it has disposed of 50% of the shares of USL. As USL develops and licences UNIX, this means that AT&T is no longer the sole controller of SVR4's fate.

In addition to its owners exerting influence, technical guidance to USL is provided by UNIX International (UI). UI was founded in 1989 as an association of over 230 UNIX suppliers dedicated to SVR4. It publishes the annual *UI Roadmap*, which outlines what the members want from future versions of SVR4 and when. In the SVR4 UNIX



Why Choose?

Between Relational and Network when db_VISTA™ III has both and now MUCH MORE

The db_VISTA™ III database development system is intended for use by the professional C applications developer. db_VISTA III is written in C and provides a complete set of sophisticated development tools that feature:

- High speed access to large or complex data
- Mainframe functionality
- Portability to any C environment
- Royalty-free run-time distribution
- Source code available in C.

db_VISTA III provides both Relational and Network model technology for programming flexibility. Retrieve a record using the relational keyed access method and all related records can be immediately available using the network database model.

db_QUERY provides SQL-like access to **db_VISTA III**, callable from within an application, **db_REVISE** allows restructuring of live databases.

db_VISTA III is fast. Access times for retrieving data are largely independent of the size of the database.

db_VISTA III is portable. Develop an application in any environment and your database access code can be ported without change to any other supported environment. Currently, **db_VISTA III** supports MSDOS, MS Windows, OS/2, Macintosh, QNX, XENIX, UNIX System V and Berkeley, Ultrix and VMS.

db_VISTA III is a product of Raima Corporation and has been sold to thousands of C programmers in over 50 countries. **db_VISTA III** is fully supported in the UK and Eire by Systemstar Ltd. Comprehensive training is available together with bespoke system development. At Systemstar we offer **C-scape 3**® screen interface library with Look and Feel™ from Liant Software Corporation to provide **db_VISTA III** programmers with a complete application development environment.

For more information about **db_VISTA III, db_QUERY, db_REVISE** and **C-scape 3** call Systemstar in Hertford on **(0992) 500919.**

FEATURES	YES	NO
Single and multi-user available	✓	
Relational B-tree indexing	✓	
Network database model	✓	
Multiple database access	✓	
Referential integrity	✓	
Transaction processing	✓	
Automatic recovery	✓	
Record and file locking	✓	
RAM resident		✓
Relational Query and report writer	✓	
Total database redesign/restructuring	✓	
C compilers: most supported	✓	
C ++ compatible	✓	
Operating Systems: VMS, Ultrix,	✓	
UNIX, BSD, SunOs, XENIX, QNX,	✓	
MSDOS, OS/2 and Macintosh	✓	
Windows 3	✓	
LANs: Netbios and Appleshare	✓	
Read and write WKS, WK1 and DBF files*	✓	
Source Code available	✓	
Training courses available	✓	
Run-time Royalties (Absolutely NOT)		✓
*using WKS Library		

NEW SHOWCASE

Object Manager Class Library
hyper-VISTA for Hypercard
4D_VISTA for Macintosh
Protected Mode for **MS-DOS**
VISUAL BASIC support for Windows
PowerCell, C spreadsheet library

CIRCLE NO. 185

SYSTEMSTAR LIMITED
1-3 PARLIAMENT SQUARE HERTFORD SG14 1EX
TELEPHONE: (0992) 500919 FACSIMILE: (0992) 554261

world, the days of 'take it or leave it' have long gone.

Structure of SVR4

SVR4 is divided into three distinct components. Figure 3 shows how these are organised in layers. The kernel schedules processes and controls their execution, manages memory and handles I/O. It also provides access to system resources via a set of system calls. Ultimately it is these systems calls which define SVR4.

The C language libraries sit on top of the kernel. These provide access to the system calls and, in addition, contain a number of frequently required services. Examples of the kinds of functions that are in these libraries are the string manipulation routines and 'stdio'; which should be familiar to most C programmers. Indeed, the raw system call interface is not very programmer friendly. The original developers of UNIX relied heavily on the philosophy of Albert Einstein: make everything as simple as possible, but not more so.

On top of the interface provided by the system calls and the C libraries are the programs and utilities which have made UNIX such a rich environment for software developers. The shells are particularly worthy of note. Unlike most other Operating Systems, the shells have no special privileges and are in fact just ordinary programs. For this reason it is often said that if you don't like those provided then you can write your own. This is akin to building your own house; anyone could, but not many people do.

SVR4 ES is really designed for use where secrets or money are involved

There are three shells in popular use. The Bourne shell, named after Steve Bourne, the author, is the standard shell. It has a syntax not dissimilar to Algol-60. Next is the C shell, which you've guessed it, has a syntax like C. I've used the C shell now for nearly 10 years, but its similarity to C escapes me. The C shell is the oldest of the three, being developed at the University of California at Berkeley which first acquired UNIX in the 1970s. Finally, there is the Korn shell. Once again, this is named after its author, David Korn. The Korn shell is the 'new kid on the block', trying to incorporate the best features of the Bourne shell and the C shell.

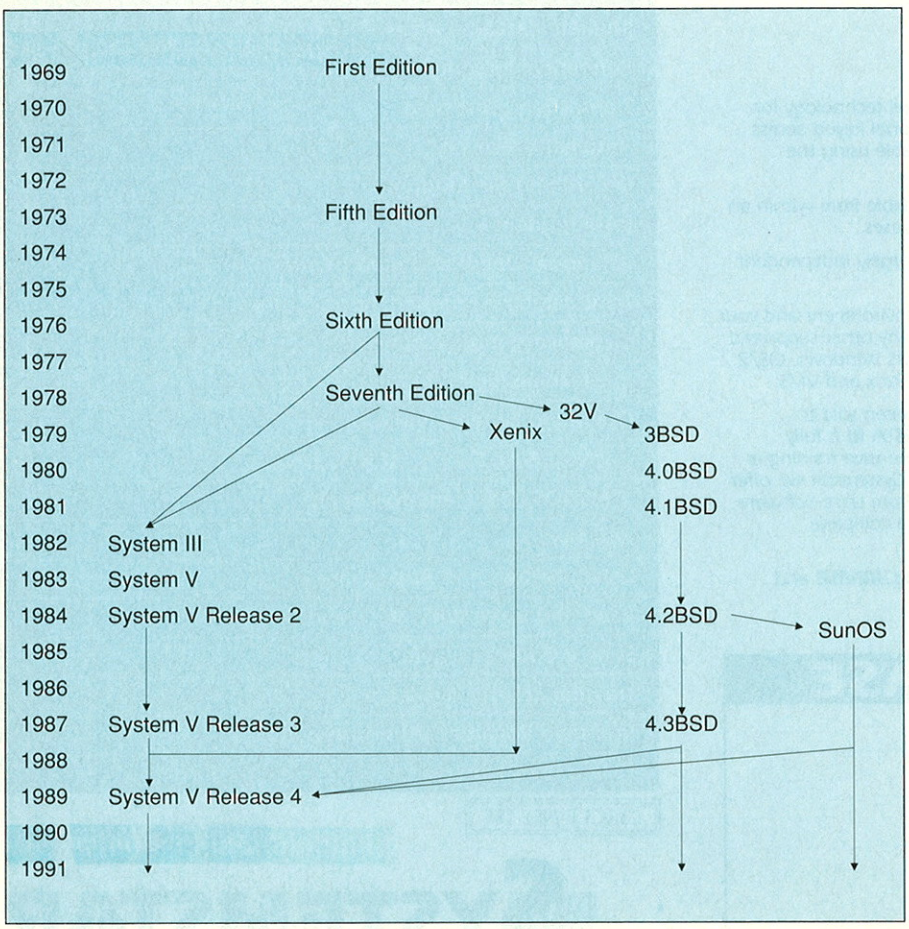


Figure 1 - UNIX Family Tree

US	Far East	Europe
AT&T	Fujitsu	ICL
Novell	NEC	Olivetti
Amdahl	III (Taiwan)	
Motorola	Oki	
Sun	Toshiba	

Figure 2 - Owners of UNIX System Laboratories

There are some wonderful utilities provided *gratis* with SVR4. The fact that they are incorporated in most other variants of UNIX mean that they tend to get ignored in the 'my UNIX is better than yours' discussions. 'Make' is one that springs immediately to mind. The concept is so simple: automating the process of compiling and linking programs using a description of how this should be accomplished. Once this description, called a *makefile*, is defined, Make ensures that the minimum amount of work is done to build the program. This idea has now spread far and wide. Make has been ported to most Operating Systems from MS-DOS to VMS.

Another freebie, which goes hand in glove with make, is the source code control system (SCCS). The SCCS utility records changes made to program code. It works as follows. When a programmer needs to change a file, he must first retrieve the file from SCCS. When his modifications are complete, he returns the file to the SCCS system, which asks for a small descriptive comment describing the change. This is recorded, along with the exact changes made. The history of change to a file can be inspected and previous versions can be returned to when the programmer, heaven forbid, makes a mistake. A really useful feature of SCCS on multi-user setups is that it can enforce the discipline of only one person modifying a file at any given time. Without this, two programmers could take a copy of a source file and modify it independently of each other. When they came to put the file back, one programmer would lose his changes. If this has ever happened to you, you will appreciate the value of SCCS.

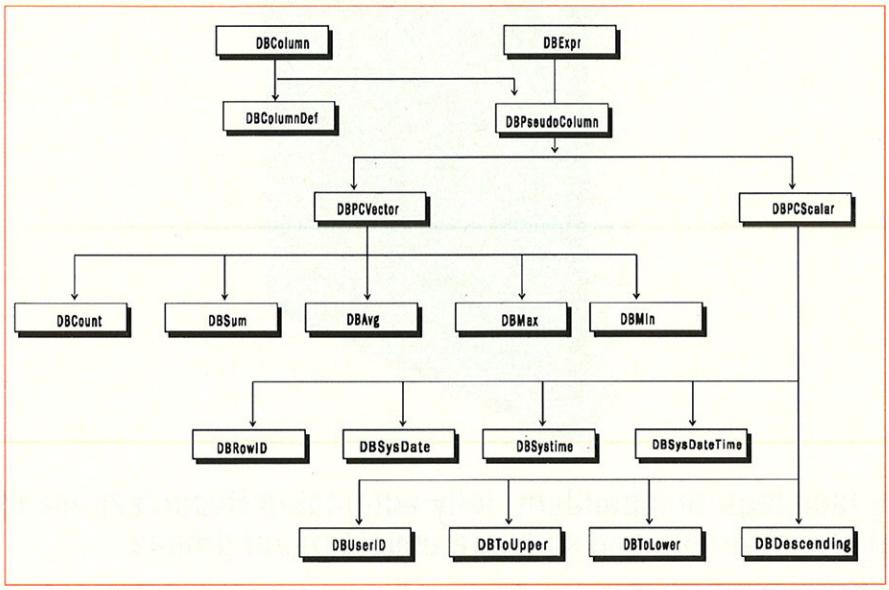
Features of SVR4

One of the goals of SVR4 was to reconcile all the different versions of UNIX. The three

CommonBase™

Database portability for C++ developers

Write portable database applications the easy way!
Use C++ and CommonBase and you can forget about the differences between APIs. Get the job done while they debate standards for "Dynamic SQL" and an SQL API interface. Harness the power of C++ and let CommonBase manage your application's data conversions and storage details across multiple platforms.
C++and CommonBase – the portable solution to information management.



A sample from the CommonBase class hierarchies

First we brought you Glockenspiel C++ – the most portable C++ compiler around.

Then we brought you Glockenspiel CommonView, the portable class library for developing Windows, PM and OSF/Motif applications.

Now we bring you the solution for portable databases – the CommonBase C++ class library.

Use C++ and CommonBase and forget about the hassles of working with different databases, operating systems and compilers. CommonBase is portable across all three.

It takes care of the differences between the various APIs and the problem of dynamically managing many cursors in Embedded SQL.

The CommonBase class interface for ISAM databases is a subset of the interface to SQL databases. So it's easy to write your application using an ISAM database and later port it to an SQL database – without facing a major rewrite.

Reusable, portable code!

It's what C++ is all about.

Specifications:

CommonBase requires Glockenspiel C++ or Borland C++.

CommonBase supports Microsoft / Sybase SQLServer, Oracle, Coromandel Integra, Coromandel ObjecTrieve, Gupta SQLBase.

CommonBase contains approximately 35 classes which are organized into separate hierarchies – there is no base "Object" class.

CommonBase documentation inclues tutorial and class library reference manual and a Quick Reference guide to classes and functions.

glockenspiel
class constructors

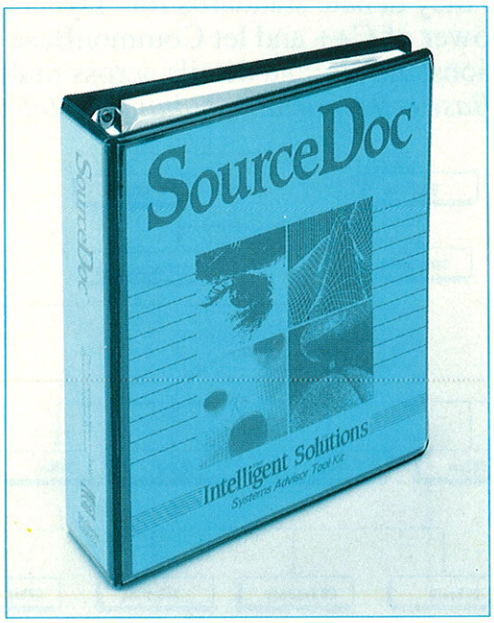
PUTTING C++ TO WORK

To order:

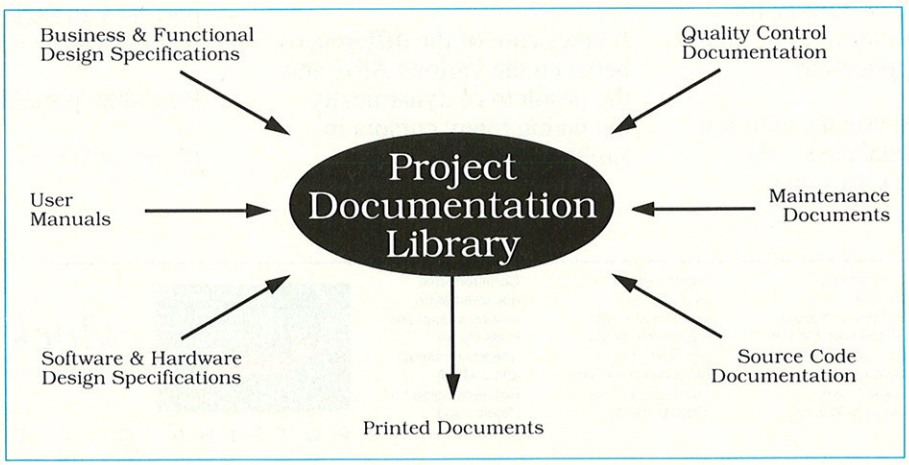
Glockenspiel, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034.
North America: Imagesoft Inc., 2 Haven Avenue, Port Washington, NY 11050. (516) 767 7839. Fax (516) 767 9067.
UK: *QA Training Ltd.* (0285) 655888. Fax (0285) 650537. **Italy:** *Inferentia,* (02) 26680568. Fax (02) 2364258.
Benelux: *Rijnhaave,* +31 (71) 218121, Fax +31 (71) 216118. ***Sweden:*** *Linsoft* (01) 3124780. Fax (01) 3152429.
France:*Microformatic,* (01) 48701900. Fax (01) 48702729. ***Germany, Switzerland, Austria:*** *PSI,* (06021) 492-0. Fax (06021) 492-112.

REVOLUTIONISE YOUR DOCUMENTATION WITH

SourceDoc™



- **SourceDoc: The language-independent, fully automated HyperText documentation tool for professional programmers and software development groups**
- **SourceDoc creates a Dynamic Access Project Documentation Library for system documentation**
- **SourceDoc is currently used by the following companies: Ashton-Tate • Compaq • Epson • Fox Software • Hayes • Hewlett Packard • IBM • Intel • Lotus • Novell • SBT Corporation • Tektron IX • Texas Instruments • Xerox**



For further information call
Readmar Systems (071) 625 5255

APPROVED DEALERS: **Grey Matter Limited**, Prigg Meadow, Ashburton, Devon TQ13 7DF, Tel: (0364) 53499 • **Roundhill Computer Systems Limited**, Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU, Tel: (0672) 84535 • **System Science**, 3-5 Cynthia Street, London N1 9JF, Tel: (071) 833 1022 • **Reflex Technology Limited**, 9 Buckingham Place, Bellfield Road, High Wycombe, Berks HP13 5HW. Tel: (0494) 465907

CIRCLE NO. 187

main systems were System V, Xenix and Berkeley UNIX (which was 4.2BSD). SVR4 combines the best features of the systems that it merged. Being compatible with Xenix brings with it the largest installed base of UNIX systems. BSD, and in particular the Sun Microsystems variant SunOS, is a very feature rich environment, TCP/IP, NFS (a distributed file system), job control (a mechanism for running multiple applications from the one terminal) and reliable signals (signals are the software equivalent of hardware interrupts, so reliability is very important).

SVR4 is not only functionally very rich, it also runs on a number of different processors. Application Binary Interfaces (ABIs) define binary portability. ABIs exist for the Intel i386/i486, Motorola MC68000, SPARC and MIPS architectures. This is very important for software developers, as it extends the possibility of a 'shrink wrap' market beyond MS-DOS and Windows 3. This is a very exciting prospect.

One of the major pieces of functionality that is a standard part of SVR4 is the X Window System and the OPEN LOOK GUI. The X Window System (X) was jointly developed at MIT by IBM and DEC. It is a *distributed* window system, meaning that the application and the display do not have to be on the same machine, but can communicate with each other using a protocol called the X protocol (surprise). SVR4 includes X and a toolkit which implements the OPEN LOOK Graphical User Interface (GUI). AT&T and Sun Microsystems together defined a standard for GUI 'look and feel' called OPEN LOOK.

OPEN LOOK borrows heavily from the work done by Xerox at the Palo Alto Research Centre (PARC), which produced such notable systems as the Xerox Star and Smalltalk. Star is not on the tip of everyone's tongue when it comes to mentioning great advances in the field of computers, but it should be. It was one of the first systems to exploit bitmapped display and mice. It was a visit to PARC by Steve Jobs which inspired the Macintosh. It is the Mac and Windows 3 that make X and OPEN LOOK so important to SVR4. It is very hard to compete for people's hearts and minds when they can see really sophisticated applications running on Macs or PCs.

SVR4 supports the OPEN LOOK 'look and feel', but competing against OPEN LOOK is OSF/Motif. Its claim to fame is that it tries to make a UNIX system look like a PC running Windows. I cannot see this as a most auspicious start, but OPEN LOOK and Motif are now slugging it out in the market place. One is going to win, and just like all those Betamax VCR owners, there are going to be a lot of unhappy customers out there when the choice is finally made. This habit of using its own foot for target practice does the UNIX market no good.

Unfortunately, introducing new technology like X is a Catch-22 situation. Software developers will not work on a system which has no market share, and to get a market share you must have applications. However, there are signs that this situation is resolving itself.

The Future

As I mentioned earlier, the UI Roadmap outlines where SVR4 will be going. The original release of SVR4 was in November 1989. During 1991 USL is making two new releases of SVR4.

UNIX Multiprocessing

Multiprocessing (MP) systems, which can connect multiple low-cost microprocessors into a high performance system, offer an attractive alternative to expensive mainframe environments. The general advantages of an MP system of course apply also to UNIX based MP systems. However, to date MP implementations have not conformed to industry standards, which means that they have negated one of the most important advantages that UNIX has to offer. Efforts to unify a number of different UNIX flavours into a standardised MP system ended in agreement between a number of companies, including UNIX System Laboratories, Intel, NCR, Unisys and Olivetti. The aim was to produce a single UNIX version for symmetrical MP (a *symmetrical* MP implies that all processors have the same capabilities, so extra processors can be added without software modification) based on UNIX System V Release 4. It is called SVR4 MP.

SVR4 MP is based on the 386 architecture, but can be ported without major changes to a variety of hardware platforms. The source code structure isolates the platform-dependent code from the platform independent code - ports will be available for the Motorola 88000 and the Intel i860 architectures in early 1992.

Target platforms for the system are machines which are built for symmetric, homogeneous, shared-memory architectures. However, it is not restricted to symmetric architectures; asymmetric architectures (where the processors are dedicated to certain tasks, eg I/O) can also be supported. The target hardware has to support interprocessor interrupt capability, and each processor in the system must be able to send an interrupt to each of the others. All interrupts may occur at a single interrupt priority, but the ability to interrupt at several levels, especially at a non-maskable level, is preferred.

In terms of performance, the ideal attainable scalability for an MP system is 100% of the first processor, ie an n processor configuration with 100% scalability will perform n times better that a single processor system. In real life this is not feasible. Across a wide range of applications, SVR4 MP adds 85% of uniprocessor throughput for up to six processors. The kernel of SVR4 is designed to execute up to 16 microprocessors. It is possible to move to larger or smaller configurations with the same operating system without any modification when new processors are added or removed.

SVR4 MP is designed to combine the extensive capabilities of SVR4 with the advantages of multiprocessor technology. Because it was built on SVR4 source code, it provides a natural path to upgrade licensees of both SVR4 and previous UNIX System V releases.

A future MP release - UNIX System V Release 4.1 ES/MP - is currently being developed. It will be built upon the new base operating system SVR4.1, supporting full symmetrical multiprocessing on small, medium and large scale systems. Designed to scale over 30 parallel processors, SVR4.1 ES/MP will address user area standards.

UNIX System V Release 4 Multi-Processor conforms to all major standards and runs on a wide range of machines. It will operate in heterogeneous environments. The system's adherence to standards and compatibility with former UNIX System V releases makes it the new standard for open multiprocessing systems.

Chris Papayianni, Technical Director, UNIX Systems Laboratories Europe.

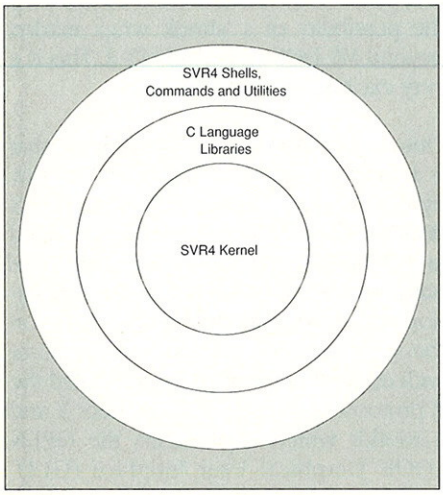


Figure 3 - SVR4 structure

SVR4 ES is the enhanced security version of SVR4. When computers and security are mentioned in the same sentence, it is usually to refer to the US Department of Defense's so-called 'Orange Book'. Using the nomenclature of the Orange Book, SVR4 ES is designed to provide C2, B1 and B2-level security. As a rough guide, B is more secure than C and B2 is more secure than B1. Of course, the more secure the system is made, the more restrictions are placed on the user. It is really designed for use where secrets or money are involved.

SVR4 MP is a parallel development to SVR4 ES. Its goal is to produce a version of SVR4 which will run on multiprocessor machines - see separate box.

Planned for next year, the final system that is on the current UI Roadmap is the SVR4 ES/MP system. As its name implies, this is a bringing together of SVR4 ES and SVR4 MP.

Conclusion

It used to be that the manuals for UNIX would fit in a briefcase. The documentation of SVR4 is approximately 18,000 pages. I suppose you could say that the manuals for SVR4 would fit in an office. This increase in size is for two reasons. SVR4 is a large and complicated system and to adequately describe it requires this level of documentation. Second, and perhaps more accurately, most users do not have access to the source code for the system, and for them, the manuals are the only source of information.

It is a testament to the original developers of UNIX, that despite all this complication, the beauty and elegance of the original ideas still shines through. Complication it may be, but oh how useful it is. X Windows and Network File Systems, RPC and screen editors; all this rich functionality makes SVR4 a fun system to use. Try it, you'll like it!

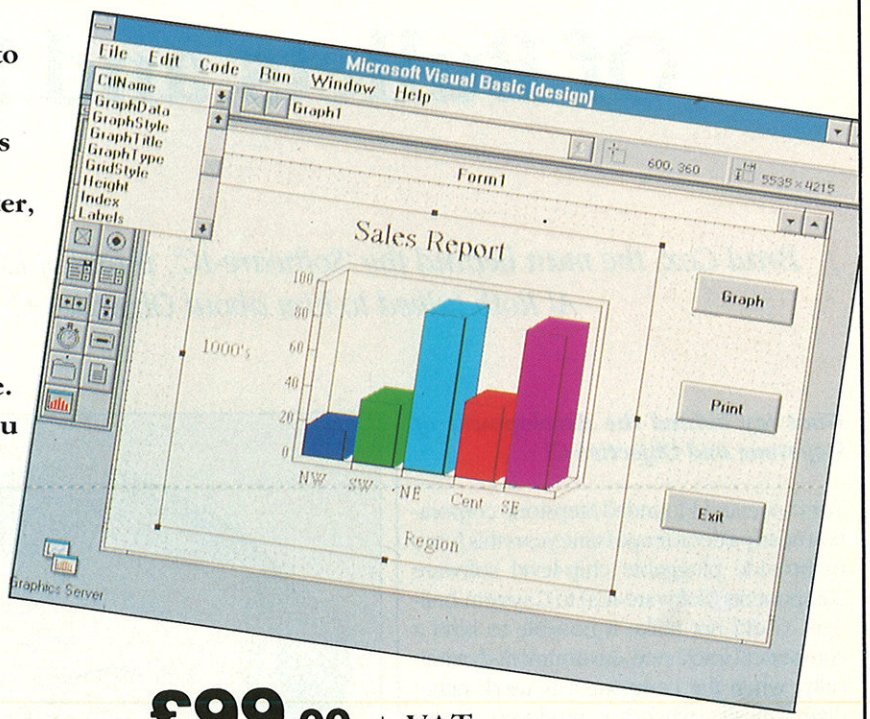
EXE

Simon Kenyon is a UNIX consultant with ICL. He has been using UNIX for 10 years now. He is chairman of the Irish Unix Users Group. He can be contacted on EUnet as simon@itc.icl.ie or by phone on +353-1-956644.

Here is the contact information for UI and USL. Both of these organisations will be able to provide you with additional information about SVR4. UNIX International (European Headquarters in Belgium) phone 010 32 2 672 3700, fax 010 32 2 672 4415. UNIX System Laboratories Europe Ltd (in London) phone 081 567 7711, fax 081 567 2420.

CHARTBUILDER
FOR VISUAL BASIC

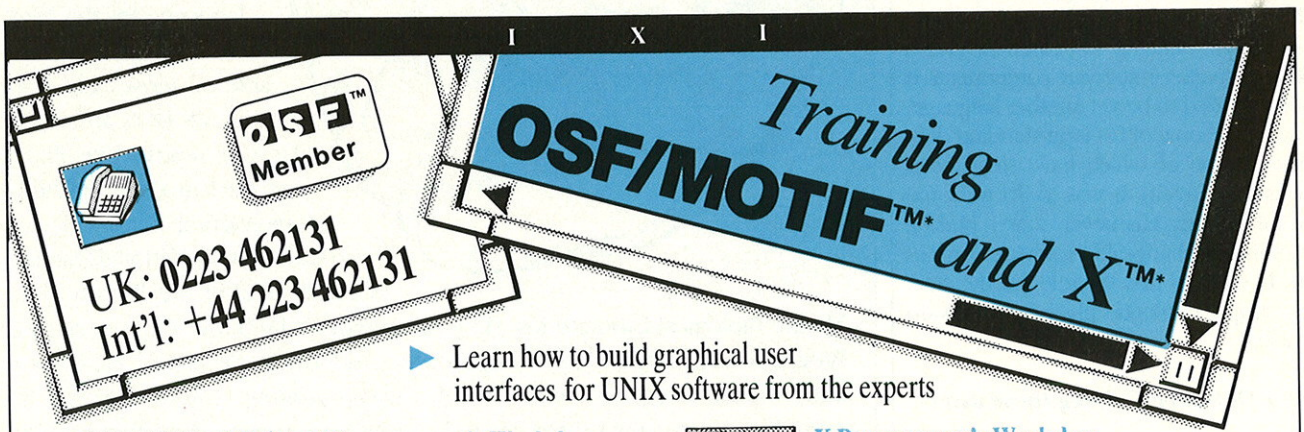
- Integrate graphs and charts directly into your Visual Basic applications.
- Manipulate your graph objects using standard Visual Basic properties, events and methods.
- Present your data as pie, bar, line, scatter, area, gantt, polar and high-low-close graphs.
- Add statistical lines and trends.
- Create high-quality hardcopy.
- Interchange images with other applications via Clipboard and Metafile.
- Pay no royalties on the applications you create.
- ChartBuilder comes complete with a royalty free run-time copy of Bits Per Second's Graphic Server, the unique charting library for Windows developers.



Bits Per Second Ltd
14 Regent Hill, Brighton BN1 3ED
Tel: (0273) 727119 Fax: (0273) 731925

£99.00 + VAT

CIRCLE NO. 189



▶ Learn how to build graphical user interfaces for UNIX software from the experts

OSF/Motif Programmer's Workshop
This popular 4 day workshop concentrates on using the Motif toolkit to build applications with OSF/Motif look and feel. Includes full course materials and extensive hands-on.

X Programmer's Workshop
Xlib, the base layer for all UNIX GUIs, is covered in detail during this 4 day hands-on workshop. Complimentary to Motif course for those writing their own widgets.

X Strategic Overview
An intensive one-day seminar on X looking at marketing, commercial and technical issues. Ideal for sales and marketing personnel or those overseeing company strategy.

On-site Training
For larger project teams, IXI provides these courses on-site. Courses can be tailored to meet your particular needs. Call Kim Fawkes at IXI Sales for more information.

I X I

IXI Limited 62–74 Burleigh Street Cambridge CB1 1OJ England Tel: +44 223 462131 FAX: +44 223 462132

OSF/Motif is a trademark of the Open Software Foundation, X is the trademark of MIT.

T R A I N I N G

CIRCLE NO. 190

Of Bullets and Muskets

Brad Cox, the man behind the 'Software-IC', is one of the best-known proponents of OOP. Al Roth talked to him about Objective-C, C++ and matters.

What lay behind the development of Stepstone and Objective-C?

Tom Love and I founded Stepstone corporation nearly a decade ago (nine years this June) to provide pluggable chip-level software components (Software-ICs) to C system-builders. C did not make it possible to bind a component into a new environment dynamically, when the component is used, rather than statically, when it is produced. So we built an enabling technology; a C front end that could support a loosely-coupled chip-level modularity/binding technology within C's gate- and block-level technologies. The goal of this extended language was to provide enabling technology for building the Objective-C System-building Environment and a multi-level market in software components. It wasn't intended to be yet another language, nor to repair some of C's long-standing deficiencies (some of which have since been addressed by ANSI). It was to do what the software industry has never done - build a market in interchangeable software components. In effect to play the role that the electronic supply catalogue plays for hardware engineers.

How is Objective-C doing these days?

Well, as a technology it is doing fine. The user-base is enthusiastic about the possibility of loosely-coupled integration of pre-fabricated software components. As a business it's not. Stepstone is essentially out of business.

Is that official knowledge?

It has been discussed out on the networks, so it's reasonably public. I don't know if it is common knowledge.

Why is Stepstone in this position?

It comes back to everything I've been saying about the software industrial revolution. Revolutions never happen quickly. Shifting the viewpoint of programmers away from languages to off-the-shelf prefabricated components is as big a problem, and as slow, as in any of the other revolutions in history.

Measuring from the time the US Armouries' customers decided to insist on interchangeable parts and allocated the funds to create a different way, to the time John Hall managed to deliver them in a pilot scale project, 25 years had elapsed. Another 25 years had elapsed before the idea spread more widely. Max Planc was unnecessarily pessimistic in his assessment, but had an eye for part of the truth. 'Change never occurs any faster than it takes the older generation to die off'.

Is it the case that you have been too far ahead of the market by trying to provide Software-ICs?

About 20 to 30 years ahead, if history is a guide.

Brad J. Cox was a co-founder and chief technical officer of The Stepstone Corporation and the originator of the Objective-C System-building Environment and many of its Software-IC libraries. Objective-C is the system programming language for the Next machine and is available on most Unix platforms, MS-DOS, and OS/2. Dr. Cox received his Ph.D. from the University of Chicago and carried out post-doctoral studies at the National Institute of Health and at the Woods Hole Marine Biological Laboratories. He is the author of the book, 'Object-oriented Programming, An Evolutionary Approach' (Addison Wesley 1986). He is presently writing a sequel, *Object-oriented System-building; A Revolutionary Approach* and is pursuing speaking and consulting engagements within his new company, Information Age Consulting.

Escape with C-scape

ELEGANT GRAPHICS AND TEXT

C-scape Features

- Graphics – Combine high-resolution graphics with text or menus.
- Object-oriented – Add features and create re-usable code modules.
- Mouse – Use any standard mouse for fast screen control.
- Portability – Write hardware independent code. Supports MSDOS, OS/2, UNIX and VMS. Auto-detects: Hercules, CGA, EGA, VGA.
- Text Editing – Create a full-featured text editor or pop-up note pad.
- Field flexibility – Create masked, protected and marked fields with complete data validation. Use time, date, money, pop-up list and many more functions – or create your own.
- Windows – Choose from pop-up, tiled, bordered and exploding windows with size and numbers limited only by RAM.
- Menus – Choose from pop-up, pull-down, 123-style or slug menus – or create your own.
- Context sensitive help – Link help messages to individual screens or fields. Cross reference messages to create hypertext-like help.
- Screen design – Build any type of screen or form with Look and Feel™ screen designer, which will then automatically convert to C.
- Screen flexibility – Call screens from files at run-time or link them.

The **C-scape**® Interface Management System frees C programmers from the tedium of coding windows, menus, data validation, help and text editing functions.

C-scape is a pleasure to use. With **C-scape**'s object oriented design you can build more flexible, more functional, more portable and unique applications – and you will enjoy doing it.

The Industry Standard.
Many thousands of programmers have given up home-grown libraries and cumbersome, inflexible products for the ease of **C-scape**. In the US, *IEEE Computer* said "**C-scape** is by far the best . . . a joy to use." *PC Magazine* chose **C-scape** to produce its Laboratory Benchmark Series 5.0 software because **C-scape** offers mouse support. Moreover, **C-scape** simultaneously combines text and graphics.

C-scape, from LIANT SOFTWARE CORPORATION, is built around an open architecture, so it can be used with data management or other C libraries. At Systemstar, we offer **C-scape** with **db_VISTA III**™ from Raima Corporation to provide a complete development environment.

To port your application from MSDOS or OS/2 to UNIX or VMS, just recompile.

Source code is included in the price and there are NO RUN-TIME ROYALTIES.

For more information about C-scape call Systemstar on (0992) 500919.

SYSTEMSTAR
L I M I T E D
1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 244

STOP SOFTWARE PIRACY

ZipitUP™

£25 Evaluation Pack Call for details

Totally Protects MSDOS Programs from Piracy, Illicit Copying and Unauthorised Use

* Easy to use - No Programming - Software Only System
Versatile & Transparent, runs on *all* 3.5"/5.25" LD/HD Formats
* Use *ZipitUP* on your PC, PC Disk Autoloader or you can bulk duplicate

Stop it happening to Your Software - Contact us **NOW!!!**
Tel: (0865) 842224 Fax: (0865) 841507

THE **data** BUSINESS Experts in Data Resources
3 Bankside, Kidlington, Oxford OX3 1JE

ZipitUP - the Best Answer There Is. . .

CIRCLE NO. 191

CopyControl

THE NEW GENERATION OF COPY PROTECTION

NO	User hassle
NO	Back-up problems
NO	Hardware add-ons or special disks needed
NO	Changes to source code required
YES	CopyControl beats ALL bit-copier Programs
YES	Floppy disks, hard disks and networks supported
YES	CopyControl is totally transparent to the user
YES	You can limit program use by no. of copies, no. of executions or date
YES	CopyControl works on all IBM compatibles
YES	Free demo disk available

For Further Information Phone or Write to:

microcosm
Microcosm Limited,
17 Cranbrook Road, Bristol BS6 7BL
Telephone: 0272-441230 Fax: 0272-427295

CIRCLE NO. 192

THE CONCEPT BEHIND OUR CASE PRODUCT.

System Architect not only has the power and flexibility to handle your most complex application, it is *easy to use* and *affordable*. As the only CASE tool to be specifically developed for MS Windows, System Architect brings you a graphics oriented, fully integrated multi-user development platform for PCs.

In addition to standard CASE tool features, System Architect offers a wealth of additional functionality.

Methodologies Supported

- SSADM
- Yourdon/DeMarco
- Ward & Mellor
- Object Oriented
- Gane & Sarson
- Information Engineering

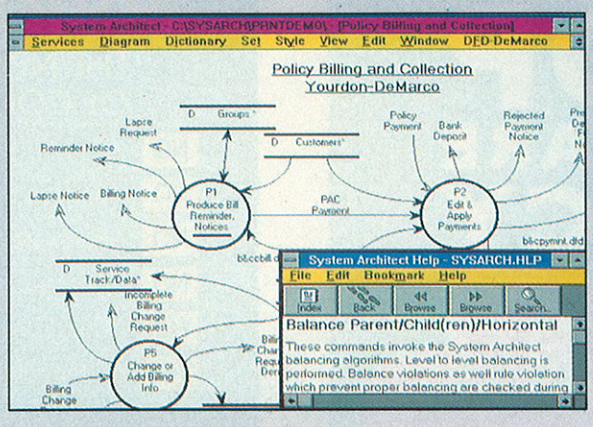
Full Data Modelling

Lifecycle Support

- Requirements Tracking
- Analysis/Design
- Prototyping
- Schema Generation
- Code Frame Generation

Benefits

- Ease of Use
- Readily Affordable
- Networking
- PC Based
- Increased Productivity



To arrange a demonstration, attend one of our regular System Architect seminars, or for further information please phone us on

Tel: (0926) 450858

Fax: (0926) 422165

Special Features

- User Definable Attributes
- Customisable Reporting System (Matrix and Tabular)
- Automated Documentation
- Import/Export/Extract/Merge
- Multiple Document Interface
- Import of External Graphics
- CRUD Matrices
- Mini-Spec Balancing
- OS/2 Support

Proven Reliability

- Over 6,500 licences installed. Users include:
  - NCR • Toshiba • Racal
  - IBM • Bull • Prudential

Training and Support

Full training and support for both the novice and advanced user. Also training courses in Structured Methods and Object Oriented approach.

Supporting IBM's AD/Cycle

MICROSOFT WINDOWS
Version 3.0 Compatible Product

Real-Techniques and Methods Limited

118-120 Warwick Street, Royal Leamington Spa,
Warwickshire, England CV32 4QY.

CIRCLE NO. 193

System Architect™

Objective-C

What is going to happen to Objective-C?

Objective-C has a life of its own and it will continue apace. The Free Software foundation has an Objective-C compiler, a consequence of NeXT's enthusiasm for loosely-coupled integration in software. There are various people, including Stepstone employees, who are determined to support it as individual business opportunities. Stepstone has been around for nearly a decade. Its revenues have never been remarkable, and it has always been driven forwards mainly by missionary zeal on our part, and oddly, on the part of our investors who saw from the beginning that the software crisis cannot be allowed to continue indefinitely. That this zeal could carry us forward for a decade is nothing short of remarkable. But at the same time, it is not something that can go on indefinitely.

So what enhancements do you see to Objective-C, in terms of functionality, or syntax?

Objective-C must acquire C++ instead of C as its substrate. C++ is a better C; a better tool for tightly coupled integration. Its weaknesses emerge when one needs a tool for loosely-coupled integration. Putting Objective-C on a state of the art C would provide compliance with the evolving C standard. And it would provide loosely-coupled object-oriented capabilities as well as the tightly-coupled capabilities of C++.

Loosen up

You believe that the object-oriented features of C++ are one of its weakest points. Why?

Consider the distinction between loosely-coupled and tightly-coupled. Hardware engineers consider this distinction when deciding to build out of pre-fabricated boards or chips, as opposed to fabricating a custom chip using gate and block level integration. The decision is about coupling. When they need a tightly-coupled solution they'll use wafer scale integration. But if they need a loosely-coupled system, perhaps so that they don't need as much skill to assemble chip-level components as gate-level components, or because they are more interested in productivity than in flexibility, they buy pre-fabricated components and simply plug them together.

Real engineers view this loosely- versus tightly-coupled distinction as an ordinary, unemotional, garden-variety, uninteresting trade-off. 'Software engineers' treat it as a religious issue. There are tightly-coupled bigots and loosely-coupled bigots because current languages don't allow flexible choice. This is why I advocate hybrid languages in which both binding options are thoroughly supported. Purist languages provide no option but bigotry, since the choice only boils down to 'which language?'. Hybrid languages provide the option of sensible engineering. Each programmer has the option of rationally asking, 'Which tool, which object-oriented level of the same hybrid language I've always been using, is the right tool for *this* job'.

What is the tension between tight and loose coupling? How does the distinction help the programmer?

A synonym for tightly-coupled is statically type-checked. Statically type-checked means that all the connections are specified in advance, hard-wired right there in the source code, so that they can be checked at compile time. Loosely coupled means pluggable. A socket must accept multiple chips, and chips must fit in various sockets, and neither must be hard-wired with knowledge of the other. It sounds better to say that you favour static type-checking than tight-coupling, but in fact they mean the same thing.

Much has been said about static and dynamic binding. My understanding is that C does static binding although we can do things late using virtual functions and pointers. Objective-C does things late unless you know that you want to do them early.

Those words have been sorely abused within object-oriented circles to imply a binary either-or situation. Common sense says early and late are with respect to a continuum. That 'early' means compile time in one context does not mean there's not earlier and later possibilities as well. C++ consistently positions itself on the earlier end of this continuum in order to obtain tight coupling. Objective-C consistently positions itself on the later end of this continuum in order to obtain loose coupling. Objective-C is focused on making it possible to build loosely-coupled software out of pre-fabricated components with tight-coupling as something you do when you must fabricate a missing component. Tight-coupling is viewed as an optimisation technique, whereas C++ has exactly the opposite set of priorities. Objective-C's shortcomings derives from C's deficiencies in fabricating tightly-coupled software components. Objective-C is a strict superset of C and C has well-known problems. This is the essential argument for Objective-C++; a loosely-coupled superstructure on top of C++, just as Objective-C was based on C.

Under what circumstances would you select one type of coupling over the other?

Loosely-coupled means changeable. Examples of places where you want to change things late is reuse of code. Here is some code that was written over here, and you want to plug it in over yonder, without recompiling it, and re-editing all the source to describe its new environment. Reuse is not C++'s strong suit, certainly not when compared to Objective-C. It is being hyped quite strongly because of the raging enthusiasm for C++ as a better C. But the reuse part is not working out as well. C++ people are beginning to realise just how grim things can get when their problem demands loose coupling.

Suppose Intel decided to sell blocks and gates (cell libraries) instead of pre-fabricated chips. Imagine the skill their customers would need in order to use cell libraries as opposed to chips. Every customer would need a silicon fabrication line. C++ is a silicon fabrication line. It does let you reuse other peoples cell libraries, but you need tremendous skill and you must adjust each cell to fit in a new chip. Objective-C is a soldering iron; a very different technology from a fabrication line. It's loosely-coupled, pre-fabricated chips (Software-ICs) can be bought of the shelf, already compiled and ready to plug in. They don't need to be changed to go in a new circuit. Its not statically type-checked, but neither are hardware chips. This is why it is so much easier to learn to use a soldering iron than a silicon fabrication line.

Inheritance

Moving on to some of the object-oriented features. C++ has multiple inheritance, Objective-C only allows strict single inheritance. How do you view multiple inheritance?

I'm not even totally happy with single inheritance. The reason comes directly from everyday experience. I look out of the window and I see trees, and birds, and rocks, and all kinds of tangible things. They're objects. Everybody agrees that they're objects. But they don't have inheritance. Inheritance played no part in creating them, but is imposed upon them externally, by the mind of each observer. Inheritance is a mental thing, a classification scheme, that is not intrinsic to the object but is imposed on it externally.

The answer is not entirely this simple. There is this other domain of experience in the tangible world that is poorly present, if at all, in software. This is the specification domain of rulers, scales and clocks, as opposed to the implementation domain of saws and hammers and programming lan-

guages. In the specification domain, one is uninterested in how objects will be implemented. One is engaged in specifying what they do. In this as yet absent domain of software specification tools, multiple inheritance has a legitimate role to play. In today's world of obsessive infatuation with implementation tools (programming languages), it is debatable whether inheritance belongs there at all, except that it is occasionally useful as a hack. Since inheritance can be useful as a hack, I decided to leave it in Objective-C. But I've been exceedingly reluctant to include more of it by providing multiple inheritance.

Inheritance is a static relationship between classes. Being static, there can be only one of them. By contrast, the tangible objects that you meet on the street are classified by every observer according to the uniquely special viewpoint of that observer. I might classify birds very differently from ornithologists, and both of us differently from those who keep birds as pets. Each has a different points of view from which to classify the same object. None of these viewpoints, these classification schemes, have anything to do with the object itself, only the point of view of each observer.

We ran into this in Stepstone. I'm not intellectualising here, but describing a real practical everyday problem that we continually faced at Stepstone. We built commercial classes using inheritance to convey a hierarchy that made sense to us as programmers. But sell it to a customer and immediately you have another view of what makes sense. How do you organise your reference manuals? Do you organise them by inheritance, so that all the array-based collections are grouped here and the list-based collections are over yonder? Or do you turn it around to fit the user's view, in which all collections are similar regardless of how the programmer built them? Regardless of whether they are array-based or list-based, here are all the things that collect entities into an ordered list. There is no way to serve both interests with a single scheme hard-wired into a programming language. It's not as simple as that.

That's why I get so annoyed with some of the multiple inheritance obsession. People are thoughtlessly taking a very rich, important and complex issue that drastically needs effective thought and effective tool-building. They're trivialising it by treating it as a whizzy feature of the latest programming language to come down the track. All this focuses on programming language technology. We obsess about this endlessly, but who is doing anything about specification technology?

In America we often link the industrial revolution with introduction of interchangeable parts in the early American musket manufacturing armouries. Although this period of history involved tremendous advances in implementation technology, by introducing powered machinery where hand-work had been used before, we often forget that the truly enabling discovery of this period was the realisation that implementation tools were necessary, but not sufficient. John Hall's crucial discovery, the reason that he succeeded where others had been failing for twenty-five years, was in the introduction and deployment of a specification tool, in his case, hardened steel inspection gauges for determining a part's compliance to specification. We have not yet made this discovery in software. When we do, I believe multiple inheritance will find its home, not in the programming language, but in the specification language.

Put CASE in its place

Will CASE address this area of specification?

To my knowledge CASE can be summed up as being a higher-level programming language than C++. The focus is still on implementing, but with bubbles and arrows instead of if's and while's. But they're all saws and hammers, they're all oriented towards churning the code out; about implementing, not specifying. Where's the tool that a customer uses to describe what he wants the code to do? Specification involves a common language that producer and consumer can use to agree on what, for example, a stack is. It is a language for stating both static and dynamic properties; ie that a stack will have methods named 'push' and 'pop' (as opposed to 'add' and 'remove'). This is a static property. A dynamic property is that if you push 1, 2, and 3, pop should return 3, 2, 1. We have no technologies for doing this today. We can't even engage in a discussion about what a stack is without first deciding what language it will be coded in.

What does OOP mean to you?

If software industrial revolution is the forest, OOP is one of the trees. The focus of my interest is the forest, and trees fit somewhere within that; a necessary, even vital, component of a larger system. When the larger system is as vitally important as this one is, the components that make it up seem less interesting. What I find interesting these days are questions like how is the human race really going to buy and sell something as ephemeral and intangible as software components. The soft-

Cultural revolution

According to Cox the software industrial revolution is as much cultural as technological. He believes that revolutions happen slowly, because of what he terms 'value rigidity' - the inability to relax the pursuit of an older good to gain a newer one. Cox believes that examples of rigidity traps are common. They include notions such as the popular position that programs must be provably correct, as opposed to merely compliant to specification within a stated tolerance, or that we should be looking for solutions that will bear fruit quickly, within a middle manager's 12-month planning horizon.

The code example below illustrates this point by highlighting the shortcomings of only allowing strict type-checking. Within the example, the type of all arguments to the `add:` method of `Set` is declared at compile-time, forcing the programmer to anticipate, in advance, every usage of the `Set` class. The alternative for Cox would be to replace the strong-type declaration of `Word*` with the anonymous type name `id`. This instructs the compiler that the `add:` method is prepared to accept instances of any class (other than types such as `int` or `float`). Cox makes the point that since type-checking and binding are now delayed until run-time, it is possible to compile `Set`, and distribute it in binary, for reuse as a commercially available software component.

```
Set* uniqueWords;
Word* currentWord;
uniqueWords = [Set new];
while (getWord(buf) != EOF) {
  currentWord = [Word str:buf];
  [uniqueWords add:currentWord];
}
printf("uniquewords = %d\n",
          [uniqueWords size]);
```

This program computes the number of unique words in a document. It turns tokens produced by the `getWord` subroutine into instances of `Word (currentWord)` and adds these to an instance of `Set (uniquewords)`, relying on the set to discard duplicates. As presented here, the application is strictly type-checked at compile time. A more flexible solution would have been to declare `currentWord` to be of type `id` rather than `Word*` to delay binding until run time.

Note that Objective-C syntax is a strict superset of C syntax. The only difference visible in this example are the message expressions enclosed in square brackets.

Put an end to software piracy!

Meet the growing family of security keys from Software Security.

Each one a specialist at enforcing your license agreement in virtually any user environment you can think of. Whether it's DOS, UNIX, Macintosh or OS/2...whether it's a single user installation or a LAN.

Simply connect the appropriate key to a single user computer, or a non-dedicated file server in a network, and you control all access to your protected application.

Users, however, won't even know it's there. The keys are transparent and won't impact software functionality or the ability to make back up copies. Normal node and LAN operations are unaffected.

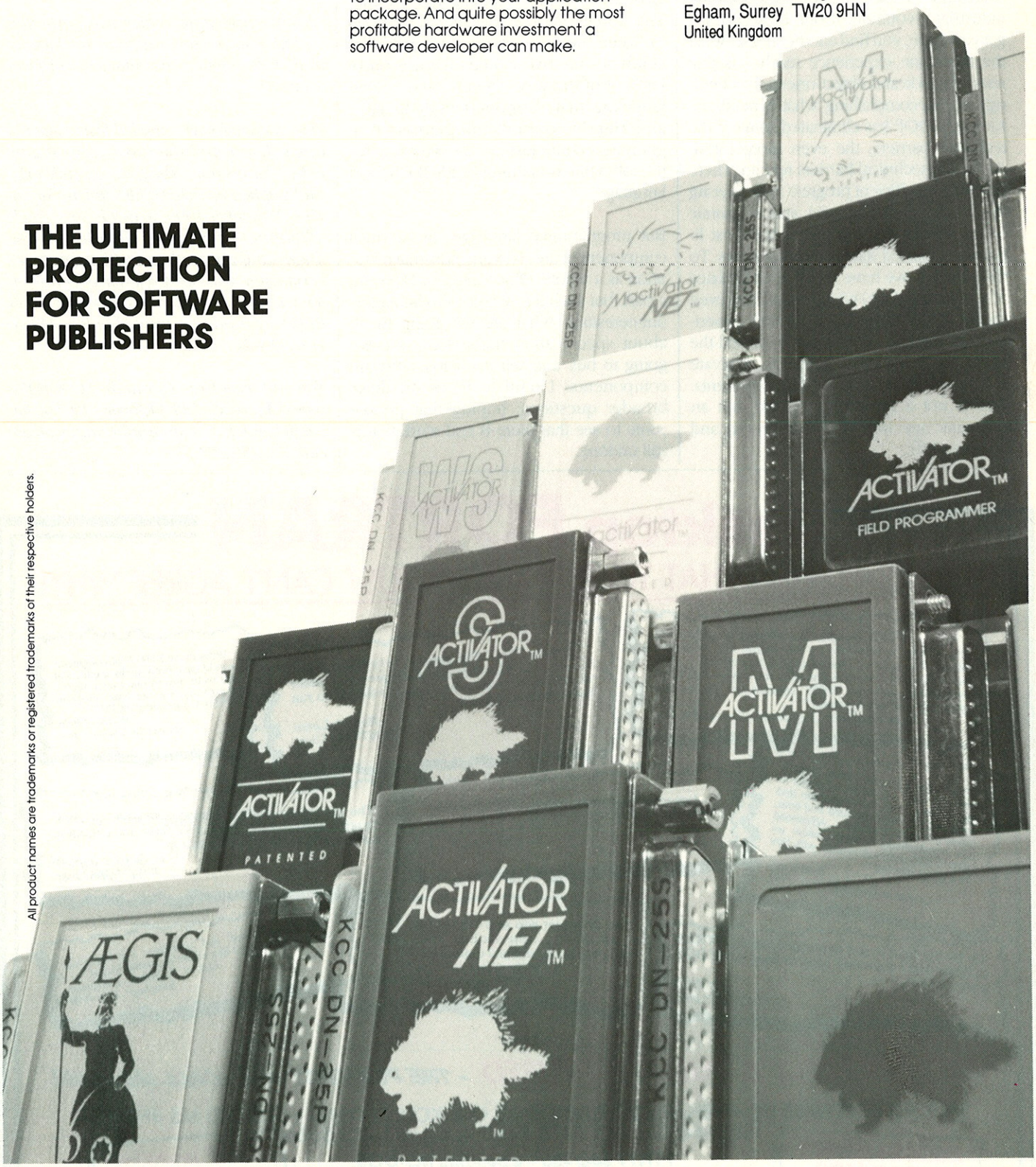
Simple. Unassuming. Ever vigilant. Easy to incorporate into your application package. And quite possibly the most profitable hardware investment a software developer can make.

To find out more, call:
Tel: 0784 430060
Fax: 0784 430050

Software Security International Ltd

21a The Precinct, High Street,
Egham, Surrey TW20 9HN
United Kingdom

THE ULTIMATE PROTECTION FOR SOFTWARE PUBLISHERS



ware industrial revolution will languish, regardless of OOP, until we devise crisp answers for such basic questions.

Where do you go next?

I'm doing precisely what I've been doing all along. Stepstone was a very ambitious initiative, but only a horse that you ride to a destination. The destination is the software industrial revolution. If one horse doesn't get you there, you find another horse. What the Stepstone experience teaches is that reusable code and object-oriented technology isn't accomplishing what it needs to. Although OOP has eliminated some of the technical barriers, the even more formidable non-technical barriers remain. There has been insufficient progress in answering the fundamental financial and economic issues of building commercially-robust libraries of trusted standard components. To the contrary, we have seen the emergence of a language (C++) which makes that even harder to accomplish. Stepstone's experience shows that no one has answered the fundamental question of how to motivate real people to build reusable components. This is not a technical question, but an economic one. How do you really buy and sell this stuff?.

C++ has a huge (AT&T) juggernaut behind it, while Objective-C has a very low profile. You have said that you don't view C++ and Objective-C in competition. However, many people do, if only because they typically are prepared to buy only one or the other. How do you view this?

Objective-C is based on C, and C++ is too. This forces people to make a decision between the two, whereas a hybrid that combines the two would take this problem out of the way. There would be one language that supported both loosely- and tightly-coupled integration. Programmers could just use the right tool for the job while remaining inside the hybrid language.

But programming languages are a small component of the software industrial revolution as a whole. Objective-C and C++ are a part, but such a tiny part as to be almost uninteresting. What are we going to do about specification tools? How are we going to buy and sell intangible software components? I tend to focus on these broader questions, hoping that people come to see that there is real work that is still undone.

Overcoming the software crisis will involve far more than amusing ourselves, as technologists, for yet another decade, with yet another round of language wars. Our consumers are getting wise to this act, and aren't likely to stand for it indefinitely.

EXE

Al Roth is a freelance writer, with particular inclinations toward artificial intelligence and OOP. He may be contacted on CIX as `alroth`.

This interview was compiled from personal discussions with Brad Cox - many thanks to him for his time. The code example originally appeared in the IEEE Software Magazine, November 1990, in Brad Cox's article 'Planning the Software Industrial Revolution', and is reproduced with the author's permission. Our other reference for this interview was the article 'There is a Silver Bullet' by Brad Cox, printed in Byte Magazine, October 1990.

You can meet Brad Cox at the SCOOP Europe conference 28th October - 1st November in London. For more information please call 071 259 2032.

JPI

the compiler people

Imagine a powerful integrated development environment common to whichever languages you choose, capable of supporting DOS, Windows or OS/2 development embodying the latest in OOP technology – it's called ***TopSpeed*** from JPI. Its modular architecture opens up new realms of choice – simply slot-in languages, source libraries, and toolkits to build a development system that meets your needs. No redundant components – what you want is what you get!

***TopSpeed* Environment** – Multi-window editor, powerful Project system, Hypertext help (environment, all languages and library), debugger, syntax checker, menu or command line driven, EMS support, pop-up calc, plus . . . plus . . . DOS or OS/2 **£59**

***TopSpeed* TechKit** – For power programming. Supports DOS DLLs, post mortem debugging, advanced overlay manager, TopSpeed assembler, .EXE file compressor, .OBJ file disassembler. **Windows 3 resource compiler**, Windows 3 run-time libraries, plus . . . plus . . . **£59**

***TopSpeed* C++** – makes C++ lean and mean Unique to JPI – TopSpeed C is the only true AT&T 2.1 C++. TopSpeed C++ gives you SmartMethod® Linking which eliminates unreferenced classes, methods and even virtual methods – a real breakthrough for OOP programming. Includes short based pointers, concurrent tasking – even with DOS, plus . . . plus . . . DOS or OS/2 **£59**

Rogue C++ Class Library **£59** DOS or OS/2

***TopSpeed* C** – *the standard is enhanced.* The only ANSI certified C. Generates compact high quality code for DOS, OS/ 2 and Windows 3. Includes run-time error checking, multi-threading, mixed memory models, links to C++, Modula-2 and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Modula-2** – the world leading implementation of Modula-2. This strongly typed and highly structured language includes type safe conversion between objects, OOP extensions with true multiple inheritance, virtual pointers, based pointers, links to C, C++ and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Pascal** – the next generation. Power-up your Pascal, convert to TopSpeed, then tune it up with the hottest compiler for DOS, Extended Dos, OS/2, or Windows. ISO 7185 conformant, Turbo to TopSpeed converter, ISO conformant arrays, dynamic strings, separate compilation units, OOP extensions plus much more . . . DOS or OS/2 **£59**

Library Source Kits available in all languages **£59**

NEW ***TopSpeed* DOS Extender**

Power-up your programming with JPI's own DOS Extender toolkit. Blast the 640K DOS barrier and supercharge your applications. Built-in disk based virtual memory management system for code *and* data, multi-tasking support using OS/2-like threads and a pre-emptive scheduler, OS/2 format DLLs supported, automatically loaded and unloaded on demand, plus . . . Royalty free. Full source code available.

Call JPI on (0234) 267500 now for your free copy of the TopSpeed 1991 Compiler Catalogue

TopSpeed products are available from:
GreyMatter 0364-53499 · System Science 071-833-1022
RTA 081-833-1022

JPI
3 The Mansards, Tavistock Street, Bedford MK40 2RX
Fax: (0234) 217094

CIRCLE NO. 196

TopSpeed® C++ TopSpeed® Modula-2 TopSpeed® Pascal TopSpeed® C

Development speed = Oregon C++

Oregon C++ is, quite simply, the fastest C++ Development System software you can buy. Consisting of a true optimising compiler (not just a C translator), a source level debugger and libraries, it sets speed records in every area. Here's how.

The optimising compiler means optimum speed

Because the optimising compiler directly generates compact object code it eliminates the translating step. The result? Faster compilation, direct debugging and faster program development. And you end up with smaller and faster applications. What's more the compiler is switch selectable for C++, ANSI C, or K&R C.

You don't waste time worrying about compatibility

Oregon C++ conforms to ANSI standards for the C++ language and supports inter-language calls to and from C, Pascal, Modula-2 and Fortran. So you get fast, easy, access to existing code modules, without wasting time rewriting or re-debugging. And Oregon C++ is totally compatible with all existing C libraries.

Fast debugging is done in the same language in which the application is written

The Oregon Debugger - ODB - debugs C++, ANSI C and K&R C in the original application language, which means you get more reliable code in less time. And because the ODB understands multiple inheritance, it can quickly display the class hierarchy.

Oregon C++ is fast and easy to use

With a choice of command line or mouse-driven window user interfaces, Oregon C++ is easy to use. In window mode you even get separate windows for the application and debugger I/O, and every window is fully scrolling so you can find the information you want in no time at all.

Fast information...

For an instant response to your request for information, call us today. Or clip the coupon for your free technical data.

Instrumatic®

The Pan-European Technology Group
Instrumatic UK Ltd.
First Ave., Globe Park, Marlow, Bucks. SL7 1YA
Tel:0628 476741 Fax: 0628 474440
Tlx:847042 IMATIC G

For more information fill in the coupon and post today!

Name

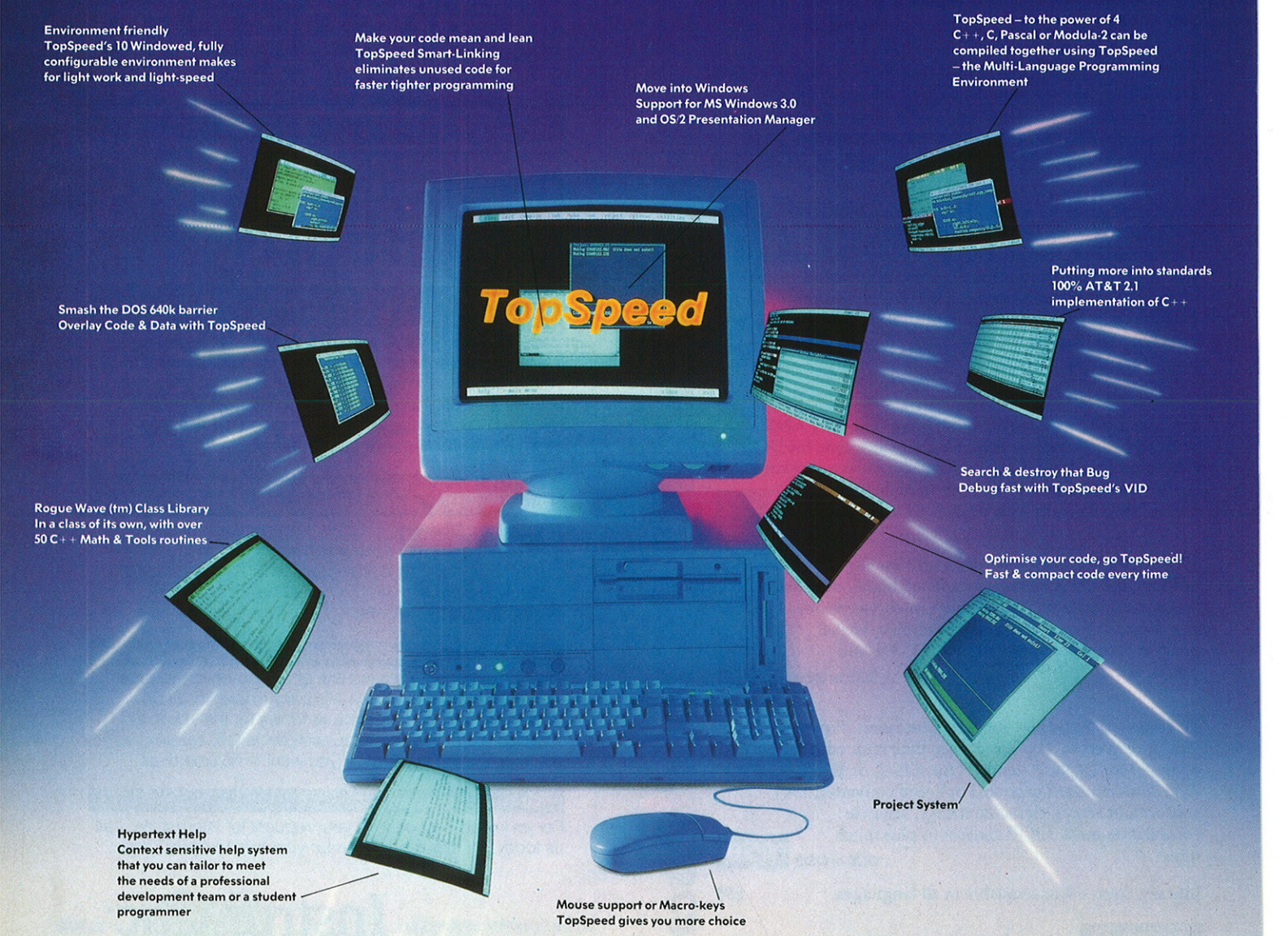
Position

Company

Postcode Telephone

CIRCLE NO. 197

If you have to get more out of your next C++ Compiler make sure there's a lot more in it



Choosing your next compiler is no mean feat. Whether you're just putting a toe in the water or about to embark on a major development project with C++, you'll need to know that the compiler you choose gives you the productive edge.

***TopSpeed* C++ packs more in**

TopSpeed C++ packs more in, whether you are developing for DOS, Windows 3 or OS/2. Develop mean-and-lean code with TopSpeed's super-optimizing code generator, Smart-Linking and Project system. Then, add the most extensive C++ class library available and you too can now harness the real power of C++.

***TopSpeed* Putting more into standards**

TopSpeed C++ not only complies with the industry standards offering 100% conformance to AT&T V2.1, but also packs in a range of extensions to take your programming into hyperspace!

***TopSpeed* just the environment for C++**

To make the most out of the C++ and the power of OOP you'll need the development environment that packs the most in. TopSpeed's unique Project system frees you to be creative whilst it manages file dependencies, compilation, assembly, linking and much more.

***TopSpeed* C++ in a family of thoroughbreds**

C++ is just one of a family of four compilers that integrate seamlessly into the TopSpeed Multi-language Development Environment. From $99 per module you can custom-build a development system that takes your programming TopSpeed!

For more information call JPI on the number below.

EUROPE
+44 (0) 234 267500

CIRCLE NO. 198

JPI
Jensen & Partners International

USA
(415) 967 3200

QuickC II - The Windowing

Microsoft QuickC for Windows is the first entirely Windows-hosted development environment for C and Windows. Paul Kemp has been exercising his mouse.

In the light of Microsoft's apotheosis of Windows, it has become increasingly paradoxical that developers of such applications in C - still the primary development language for Windows - have been confined to the DOS box for compiling, linking and debugging. If you wanted to develop applications *for* Windows *in* Windows you had to plump for either Borland's Turbo Pascal for Windows or, more recently, Microsoft's Visual Basic. Not any more, QuickC for Windows (QC/Win) is billed as 'the easiest, fastest way to develop Windows applications in C'. As with the latest Borland and Zortech C++ compilers, QC/Win does not require the Windows SDK and contains everything you need (and more) to develop full-blown Windows applications in C. The hardware used in this article was as follows: a 25 MHz, 386-based PC with 4 MB RAM.

What's in the box

More to the point, what isn't in the box. QC/Win comes with a hernia-inducing 2,000 pages of documentation, so have a wheelbarrow handy to take delivery. I don't know how swish the published versions of the manuals will be since, at the time of writing, they were still being printed. There are four manuals in the set: *Development Environment Guide; C Through Windows* (a C and Windows programming guide); *Toolkit* (which documents the Dialog Editor, Image Editor and QuickCase:W, along with compile and link options); and *Windows Programming Reference* (a hefty 1,000 pages on the Windows API). Most of the printed documentation is also in online form, although I couldn't get online help for either makefile or resource script statements.

This plethora of documentation is welcome, especially since Microsoft glibly claims that 'programmers do not need to be familiar with either C or Windows programming to use the QuickC environment'. To suggest that programming Windows in C might be that easy is fanciful, and good example programs are a little thin on the ground. However, there's more than enough rough-age here to chew on for starters.

Getting started

In keeping with the software itself, the install program is a Windows application, and the setup procedure is quite routine, if a little long-winded. After about 30 minutes I had completed a full installation with all memory models catered for, and was relieved of disk space to the tune of 9.5 MB. A cute feature of the install program is that it displays the amount of disk space it's going to require if installation were to proceed with the current option settings. The QCWIN directory tree created is refreshingly simple, comprising just \BIN, \INCLUDE, \LIB and \SOURCE subdirectories. A minor modification to AUTOEXEC.BAT to add `C:\QCWIN\BIN` to the PATH and you're ready to go (paths to the \INCLUDE and \LIB subdirectories are maintained internally by QC/Win and can be extended to utilise third-party C libraries).

What you're left with is a 'Microsoft QC/Win' group in the Windows Program Manager, displaying icons for the four main software components that make up the QuickC for Windows development system. These are:

- Dialog Editor
- Image Editor

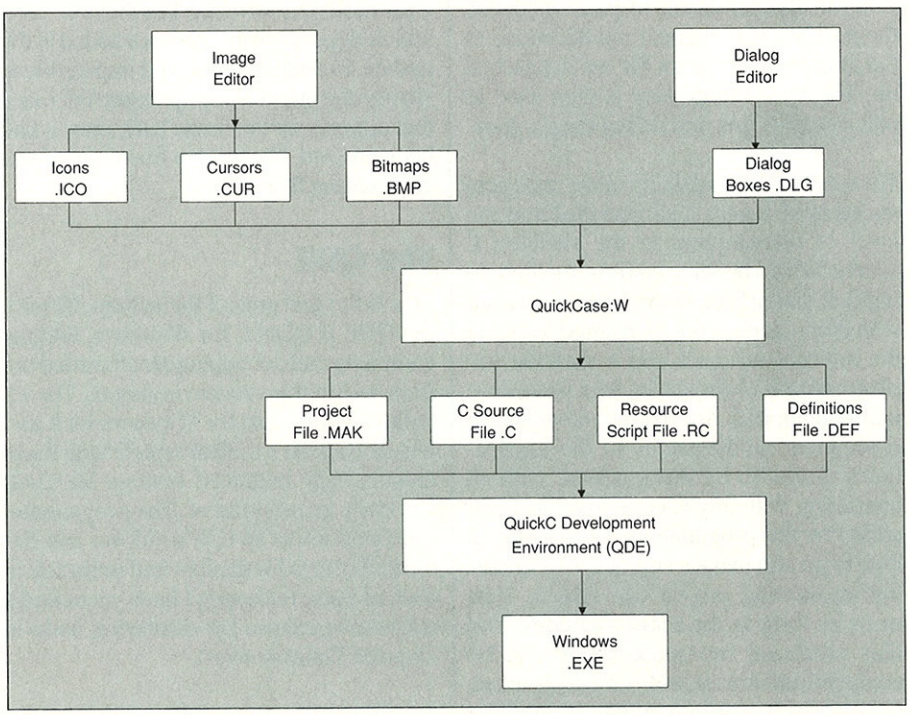


Figure 1 - Building a Windows Application with QC/Win

- QuickCase:W
- QuickC Development Environment (QDE)

The Dialog Editor is a revamped version of the one supplied with the Windows 3.0 SDK and, I am reliably informed, is identical to the one that will ship with the Windows 3.1 SDK. New features include a control Toolbox à la Visual Basic, a test mode, and facilities to force alignment and sizing of controls in the dialog box. The Image Editor is quite simply a yuppy name for the Paint program supplied with the Windows 3.0 SDK and can be used to create bitmap, cursor and icon resources.

QuickCase:W

Licensed from CaseWorks, QuickCase:W is a cut-down version of the company's Case:W code generator for Windows. Figure 1 shows how the tool fits into the QC/Win development cycle. Bitmap, cursor, icon and dialog resources are first created using the Dialog and Image editors. QuickCase:W is then used to design interactively the application's main window and menu bar. Attributes such as initial size and position, background colour and style are specified. Menu items can be linked to dialog boxes or application code (Figure 2). Selecting the *Generate* option causes QuickCase:W to digest all the information it's been given and give birth to a set of files (.MAK, .C, .RC, .DEF and .H) that implement the design. In addition, QuickCase:W maintains a binary (.WIN) file which describes the interface design - so from here on in, all updates should be performed with the tool because manual modifications to the [.RC] resource script file are not picked up. The QDE component is then used to edit, compile, link and debug the program.

The quality and readability of the code produced is of a very high standard and the level of commenting in the skeleton C source file can be selected (low, medium or high). If dialog boxes have been associated with menu items, code is inserted to create the relevant resource and a skeleton implementation of the dialog box procedure is also generated. An *Update* feature allows subsequent modifications to the application's design to be incorporated, without interfering with any C code that has been added by the programmer. This is a good time to go and make a cup of tea, as updating an existing project can take at least twice as long as the initial *Generate*. The files produced by QuickCase:W can be compiled and linked, without modification, to produce a working Windows application. For C programmers new to Windows, it is a great confidence booster to be able to get a simple application up and running so quickly and easily.

There are, however, a few points worth noting. First of all, QuickCase:W is a significantly less substantial beast than its grown-up cousins from CaseWorks (Case:W comes in two editions - Standard and Corporate). There is no support for child windows and no way to link dialog controls with application variables and user logic. It would also be useful if one could specify that a dialog box procedure be implemented in a separate C source file, since QuickCase:W creates only a single .C file and there are no facilities to manage multiple source files.

The linkage of dialog boxes to menu items is performed at the .DLG file level, so each dialog box must be saved to a separate file in the Dialog Editor. I found this rather cumbersome for two reasons. First, many applications use dozens of dialog boxes. Consequently support files proliferate because for every dialog resource in a program, three files are created - .H, .DLG and .RES (since QuickCase:W demands that a header file has the same base name as its associated .DLG file, and the Dialog Editor creates a .RES file for its own use). Secondly, because each dialog has its own header file, it becomes difficult to keep track of the various control IDs used in an application.

I also noticed that the project makefile (.MAK) was not updated to cater for dialog boxes that were linked into the application after the primary *Generate*. All other files were modified correctly, but the new .DLG and .H dependencies were not added in the update. In addition, it proved impossible to change the title of a popup menu (the name that appears on the menu bar), once it had been entered. This is very frustrating if you get it wrong first time!

The QDE

This is the real guts of the system. Indeed, the QDE *is* QuickC for Windows. Editing, compiling and debugging are all carried out from within this one environment. The interface has a Word for Windows look and feel, and sports a toolbar replete with those enigmatically engraved buttons Microsoft has lately grown fond of. The programmer can open up to 16 edit windows and has the ability to tailor the font and point size in each of these (although I have yet to find a reasonable excuse for displaying code in 24-point Times Roman).

A really useful feature of the editor is 'syntax colouring'. If it is enabled, the editor parses code that is typed in and highlights various C language elements (such as keywords, numbers, strings and comments) in different colours. The colours used are fully configurable so that, for example, one might choose to simply colour comments in green and keywords in blue. I found that comment-highlighting was particularly helpful in enhancing the readability of source code and in the early detection of 'missing */' errors.

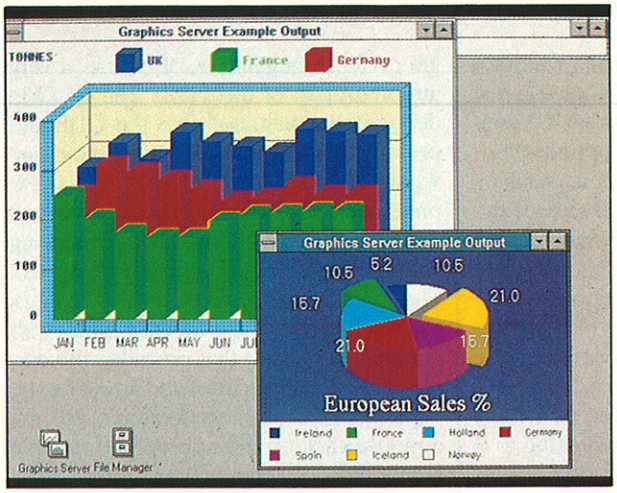
Although not a member of the 'Real Programmers Don't Use GUI Editors' club, I have to confess that I found performance a bit sluggish, especially when scrolling through text. I know that Microsoft is working on this so I shan't harp on. Basic facilities such as Search/Replace and Undo/Redo are provided, but there is no way to create and assign keyboard macros or write C-language extensions (as can be done in the C V6.0 PWB). Improvements in the speed and functionality of the QC/Win editor will certainly be required before it can compare favourably with the PWB. It was also annoying that the editor did not support the selection of text by column. But these are relatively minor grumbles compared with the advantages that an integrated environment offers.

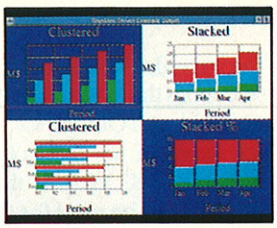
Compiling

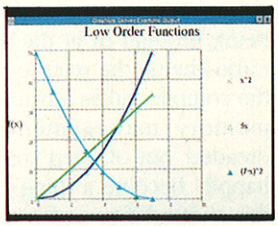
Application development in the QC/Win environment is project-based, with a project essentially defined by the contents of its makefile (.MAK). As with the Programmer's WorkBench (PWB) of C V6.0, this is built by adding support file dependencies in a dialog box. This facility is a joy to use for anyone (like me) accustomed to hand coding makefiles. Compiler and linker options can be specified in another dialog box with corresponding 'release' and 'debug' versions of the switch settings. QC/Win can be used to create both DOS and Windows .EXE files, Windows DLLs and something called a QuickWin application. The latter is a standard text-mode DOS application that has been relinked with the QuickWin library. This process replaces the DOS run-time library with one that is Windows-aware. Standard character-based I/O functions (such as `printf()` and `scanf()`) are translated into Windows API calls. What you end up with is a primitive Windows front-end to the application and a child window for Stdin/Stdout/Stderr. QuickWin provides an easy way to move simple text-mode programs to Windows where they can use the clipboard and all of the memory available to Windows, although they are, of course, far from true GUI applications. There is full support for the standard DOS run-time libraries, as with previous versions of QuickC.

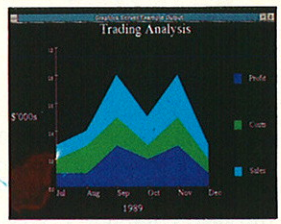
Bring Graphs and Charts to your Windows Applications Licence-Free with Graphics Server™

Introducing the **Graphics Server Software Development Kit (SDK)** from Bits Per Second. The most cost-effective, time-efficient way to integrate full-colour graphs and charts into your Windows applications. Here's why.









Client-Server Model

Graphics Server is a Windows server process which responds to multiple, concurrent requests for interactive graphing and charting functions from client applications. It enables graphs and charts to be incorporated into any Windows application having a Dynamic Link Library (DLL) or Dynamic Data Exchange (DDE) capability. This includes C programs, Superbase, SQL Windows, ToolBook, Word for Windows and many others. In fact, virtually all Windows applications!

Multiple Choice

Graphics Server lets you choose from 16 graph types, including 2-D and 3-D bar (simple, stacked and clustered), 2-D and 3-D pie, area, Gantt, scatter, polar, log-log/log-linear, line, high-low-close, X-Y and polar, plus a variety of statistical functions.

Ease of Programming

Graphics Server, through its multi-level interface, provides the level of control **you** want. In as few as **eight** calls you can open a new graph window, define titles, legends and labels, transfer data, draw a chart, and output the chart to the clipboard or printer. Graphics Server takes care of all the details; scaling the graph to match the data, matching fonts, text size and position to the graph, even resizing and repainting the graph window when necessary! You'll save development time. And money.

Alternatively, if you want full control yourself, Graphics Server provides over a **hundred** functions to help you design, create and manage virtually any kind of graphically drawn image. Multiple graphs, multiple windows, user interaction via hot regions. All these and more!

Full Support

The Graphics Server SDK gives you all the tools, documentation and examples you'll need to make immediate and effective use of Graphics Server. Within hours your applications will be transformed by the inclusion of presentation-quality graphics.

Telephone/fax support and a bulletin board service are provided for all registered users, and for power users requiring an even higher level of service and security there's a special VIP Support Programme.

No Licences!

The Graphics Server SDK costs £275 + VAT. No further licence fees to pay.

To place an order, request a demo disk, or obtain additional information, call . . .

All product names are trademarks of their respective holders.

BITS PER SECOND

Bits Per Second Ltd
14 Regent Hill
Brighton BN1 3ED
Tel (0273) 727119
Fax (0273) 731925

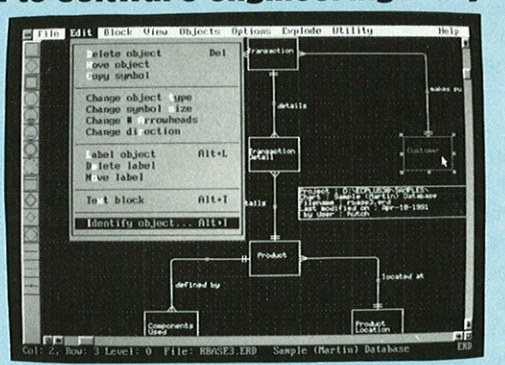
CIRCLE NO. 199

EasyCASE™ Plus

NEW! VERSION 3.0 Call for upgrade details.

The affordable approach to software engineering... *only* $495

Finally, there's a CASE tool that won't get in the way of your creativity... A tool that makes structured analysis, structured design and data modeling as easy as working with any other tool on your PC - EasyCASE Plus! Using EasyCASE Plus' new, easy to use graphical user interface (GUI), you'll be creating and editing charts, linking them, and building your data dictionary in no time. As well as being easy to use and easy to learn, EasyCASE Plus is easy on your budget! Ask any user. They'll tell you it's the best buy for your PC based CASE tool needs. Discover why over 4,000 software professionals use EasyCASE Plus and how you can join them!



"EasyCASE Plus is a well designed, low priced tool that is easy to learn and provides excellent diagramming capabilities... EasyCASE Plus is an excellent investment."

COMPUTER LANGUAGE PRODUCTIVITY AWARD 1990

Features:

- IBM SAA/CUA compliant graphical user interface (GUI)
- Extensive diagram editing features
- Integrated dBASE III compatible data dictionary
- Integrated dictionary manager, reports manager, process editor
- Hierarchical chart linking & process decomposition
- Record and element definitions
- Extensive printer, plotter and desktop publishing support
- Data dictionary import, export, and merge
- On-line help
- Comprehensive documentation with tutorial
- Access to your database, word processor, DOS, etc.
- Integrated diagram analysis (optional)

Requirements:

Runs on: IBM PC or PS/2 (AT recommended), DOS 3.1 or higher, EGA/VGA color, mouse, 640 K RAM (500 K free), 1 MB EMS recommended, math co-processor supported. Printers/Plotters Supported: Epson FX & LQ, IBM Graphics & Proprinter X24, HP QuietJet, DeskJet, & LaserJet, HP Plotters, PostScript.

EasyCASE Professional $649
(includes integrated DFD level balancing and data dictionary/diagram analysis)

Methods:

- Yourdon/DeMarco
- Gane & Sarson
- Ward-Mellor/Hatley
- Yourdon/Constantine
- Martin
- Chen, Bachman

Diagram Types:

- Data Flow Diagrams (DFDs)
- Structure Charts
- State Transition Diagrams
- Entity Relationship (ERDs)
- Data Model Diagrams
- Transformation Schema (real-time DFDs)

Evergreen CASE Tools

16650 NE 79th Street
Suite 200
Redmond, WA 98052
USA
FAX: (206) 883-7676

Call today for a brochure!
Tel: (206) 881-5149

See us at Caseworld Booth #1421

CIRCLE NO. 200

The compiler is implemented as a Windows DLL and is the same compiler engine as that supplied with QuickC V2.5. As such, it does not perform all the code optimisations available in C V6.0. If this is a problem, QC/Win can be used for development and debugging, with the professional compiler being brought in at the eleventh hour to turbo-charge the release version. Because the compiler takes advantage of Windows memory management routines, the dreaded 'out of heap space' message has happily become a thing of the past (bear this in mind if you plan on switching to C V6.0 for final compilation - it may rear its ugly head again!).

Building a project's executable in the QDE is only a mouse-click away, in fact it's kicked off by one of those mysterious icons on the toolbar. Windows versions of the resource compiler, C compiler and linker are invoked (along with IMPLIB if you're making a DLL) and a window pops up to keep you informed on how it's getting along. Compilation errors are written to an 'Error' window and can be traced back to the offending source statement by double-clicking.

I was pleasantly surprised with the performance of all these utilities under Windows and found it only marginally slower to complete a project rebuild in QC/Win compared to C V6.0 + SDK. That's not to say that it is speedy *per se* - I know that some non-Microsoft compilers are reported to be considerably more fleet of foot. On further investigation I found that it was indeed beaten by both Borland C++ V2.0 and Zortech C++ V3.0, with the Zortech compiler coming out on top. However, I was quite happy to sacrifice a little expedition for the ease of use offered by the QC/Win environment.

A good test of compiler robustness is to throw it a meaty chunk of production code and see how well it fares. This has been done at graphics software house Bits Per Second with a 70,000-line Windows program, comprising eight C source modules. Apparently the QC/Win compiler didn't even get indigestion, although it did reveal that PWB NMAKE files are not 100% compatible with QC/Win's project (.MAK) files - it is probably easiest to use the QDE to recreate, from scratch, the makefile of an existing application.

Debugging

A major selling point of QuickC for Windows is that it allows the programmer to remain in the Windows environment throughout the application development life cycle. Debugging is, of course, a very important part of this cycle. The QC/Win debugger is integrated into the QDE and provides all the basic functionality one would expect. Local variables, watch expressions and CPU registers can be viewed and modified in MDI child windows along with the executing source code (Figure 3). There is also a facility to view the program's call stack - a list of the function calls (with parameters passed) that preceded the current statement. Structures and arrays can be expanded, and pointers dereferenced with a mouse-click on the appropriate variable name. Different types of breakpoint may be set for halting program execution at a specific location, changed expression and at a Windows procedure when a particular message or instance of a message group is received (eg one could choose to trap only `WM_MOUSEMOVE` messages or widen the net to catch *all* mouse messages).

The preliminary documentation available, at the time of writing, seemed to indicate that it was possible to debug text-mode DOS applications with the QC/Win debugger - this is not the case. Windows .EXEs, DLLs and QuickWin programs can be debugged in the Windows environment,

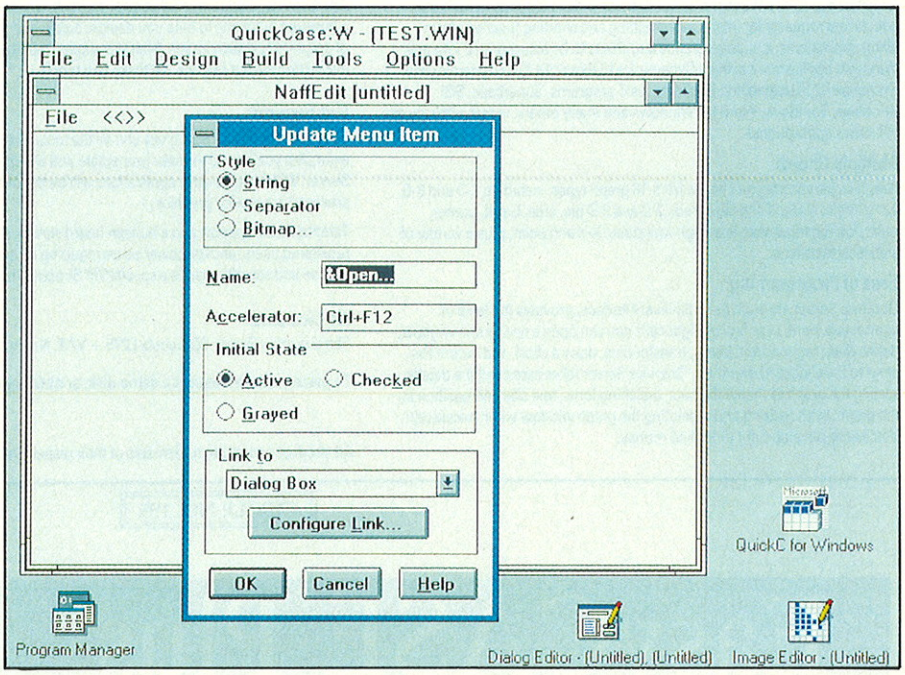


Figure 2 - Linking dialogs in QuickCase:W

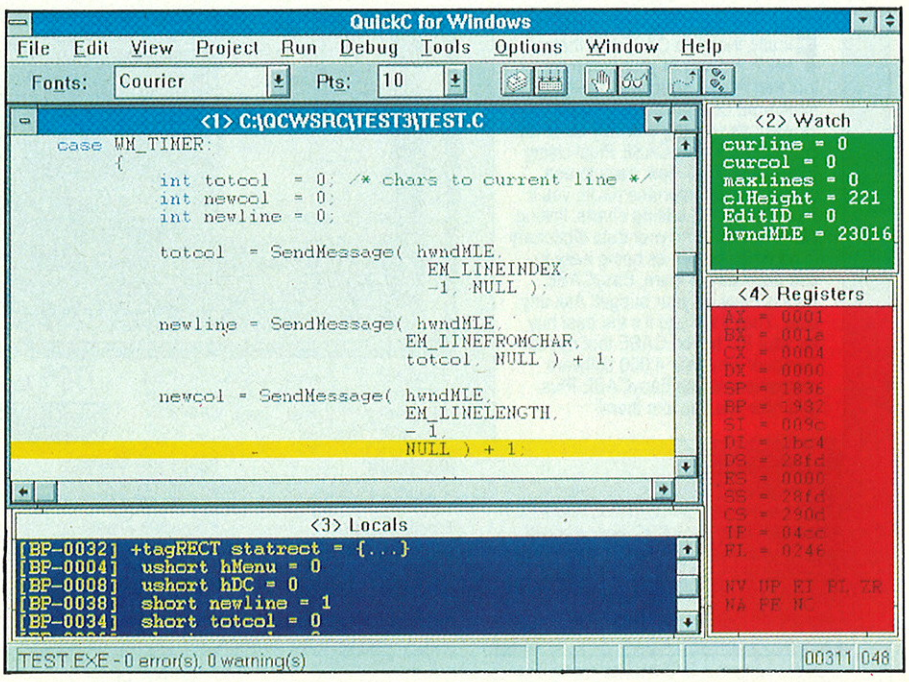


Figure 3 - Debugging in the QuickC Development Environment (QDE)

Whatever The Multiport Problem, IMPAC IS THE SOLUTION

8-64 Intelligent RS232 Ports for
UNIX, XENIX, OS/2,
DOS, CDOS, MDOS

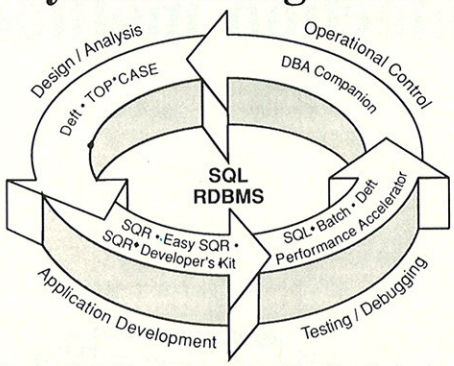
- **Easy to Install**
- **High Performance**
- **Proven Reliability**

Call SCL on 0737 762200
or Fax 0737 768472

IMPAC – FIT IT AND FORGET IT!

CIRCLE NO. 201

SQL Solutions "THE" Systems Integrators



SQL Solutions is the first single source vendor of professional services and SQL productivity tools to provide complete SQL Systems Integration.

We offer:

- **RDBMS independent Productivity tools to address each phase of the applications development lifecycle.**
- **Unparalleled expertise in the design and implementation of applications in all major RDBMS's.**
- **Gateway products to enable realtional databases to co-exist.**

To see our full range of products visit us on stand 17a at the Software Tools exhibition

SQL Solutions (UK) Ltd
Kennet House, Waterside Park
Sweetwell Road
Bracknell, Berks, RG12 1HH *Tel: 0344 714053* *Fax: 0344 714056*

CIRCLE NO. 203

High Performance PC X Servers
Get X Workstation performance and resolution from your PC – ring

0272 – 570516

Computer Resource Management Ltd
24 Sidelands Road, Downend, Bristol BS16 2TS
Fax 0272 – 575825

CIRCLE NO. 202

F77L–EM/32 &Lahey Ergo OS/386

Port mainframe programs as large as 96MB to 386/486's with this 32-bit DOS-Extender compiler. The winner of PC Magazine's 1988 Technical Excellence Award just got better. New Version 4.0 includes: Programming Tools, Popular Fortran 90 features, Virtual Memory Support, DESQView support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 and OS/386. *£845.00 plus VAT*

F77L

The fastest real-mode compiler available. F77L can take advantage of your 386 PC by generating 32-bit instructions. New Version 5.0 includes: Fortran 90 features, Weitek support, and Video Graphics. *£375.00 plus VAT*

Lahey Personal Fortran 77

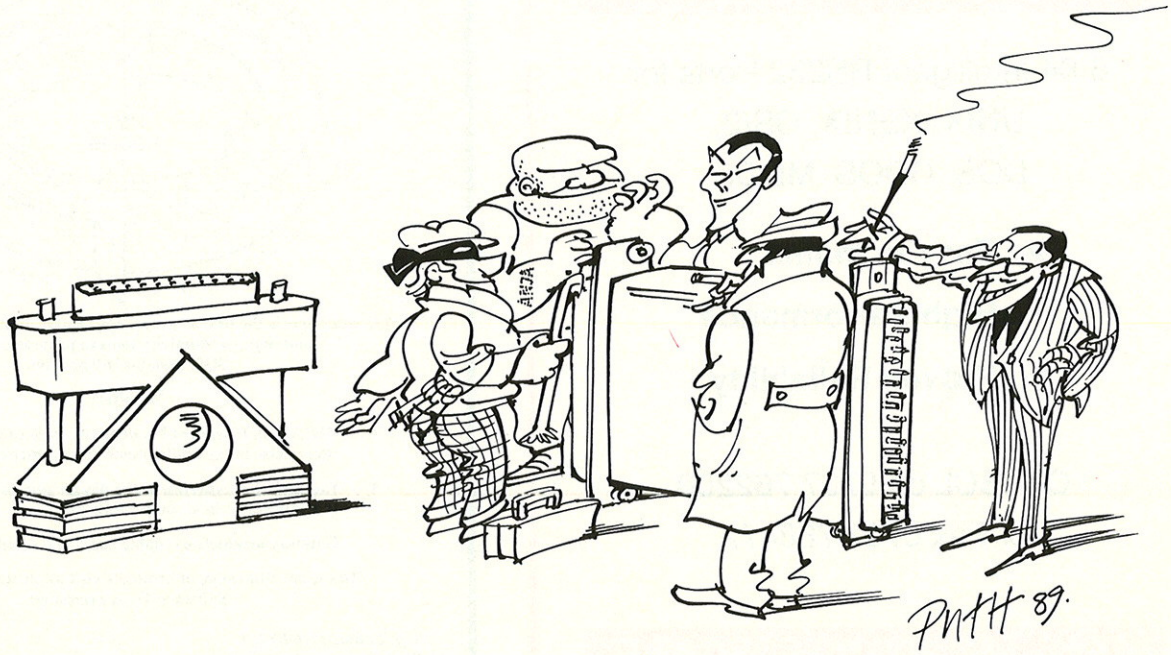
Version 3.0: Full ANSI 77, Editor, Debugger, Linker, Library Manager, Microsoft and Borland C interfaces. A great learning tool at an unbeatable price. *£79.00 plus VAT.*

System Science

3-5 Cynthia Street,
London N1 9JF
Tel: (071) 833 1022
Fax: 081 837 6411

CIRCLE NO. 204

5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E.®

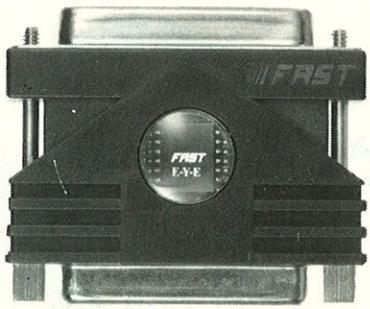


What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



Hardlock E-Y-E programmable, algorithmic response and memory option – all in one.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 205

but not DOS applications. The latter must be converted to QuickWin applications, through relinking, or debugged with CodeView, text-mode QuickC, or other DOS debugger.

The debugging interface is not pretty but functional. I found it worthwhile spending some time juggling with window sizes and positions to achieve an optimum debugging setup. Thankfully this 'workspace' can be saved and reloaded whenever it is required. Indeed, up to three workspace configurations can be saved for each project (eg edit, debug and build). On saving a workspace, QC/Win documents the screen setup including the type of windows open, their size and position. Microsoft could perhaps take a few tips on interface design from Logitech's superb MultiScope debugger for OS/2 PM (although the PM front-end can only be used to debug non-PM programs). In addition, the lack of assembly-level debugging facilities might prove a serious shortcoming for hard-core Windows programmers. To be fair however, QuickC for Windows is aimed at programmers new to Windows, and CodeView-compatible .EXEs can be generated in QC/Win for those wishing to delve a little deeper than the QC/Win debugger will allow.

To conclude

There are many good things about QuickC for Windows. Microsoft is to be commended on its introduction of a Windows-hosted C compiler, and a development system which allows programmers to build and debug Windows applications with tools that take advantage of the graphical environment. All this certainly represents a step in the right direction, but what we need is a giant leap. Visual Basic was revolutionary because it allowed novice programmers to develop quite sophisticated Windows applications with relative ease. VB's high-level object-orientation enables programmers to deal directly with graphical objects in an intuitive and encapsulated manner. However, the product does have limitations and, I suspect, most professional Windows developers would prefer not to use BASIC. QC/Win is not 'Visual C', nor does it address the real problems associated with developing serious GUI applications. The programmer still has to deal with the underlying complexity of an API without a whiff of object-orientation. Despite the benefits of a code generator and integrated development environment, writing Windows applications using the raw, C-language API remains a lengthy and arduous business.

With QuickC for Windows you undeniably get a lot for your money and I would highly recommend it to anyone wishing to dabble with writing Windows apps in C. As an indicator of what is to come, I am encouraged by the release of QC/Win, but what I want is 'Visual C++' and a professional 'C++ for Windows'. Microsoft UK Head of Languages, Andrew King, assures me that these products are on their way, but how long do we have to wait?

EXE

QuickC for Windows should be available in early October and will cost £139. Existing users of QuickC 2.x can upgrade for £79. A pack of SDK utilities (not supplied with QC/Win) comprising the Windows debugging kernel, CodeView debugger, Spy, HeapWalker, Shaker and the Windows Help compiler can be purchased for £100. Microsoft is on 0734 500741.

3D Computer Graphics - The Warnock Algorithm

The overriding goal of computer graphics has been to devise algorithms with which to generate more realistic computer imagery. Graeme Webster explains a classic approach, first suggested by John Warnock.

Previous articles in this series have presented a basic graphics library which can be used to exploit the high resolution 256 colour modes of super-VGA graphics cards and the essential algebra and routines for 3D manipulations and perspective transformations. The time has now arrived when a start must be made to tackle the classical hidden line and hidden surface problems of computer graphics. The routines presented so far will draw lines corresponding to all the edges of an object. For a 2D object, that probably does not matter. For a 3D object the resulting image does not look solid but rather looks like a wire framework, hence the description *wire-frame* image. Wire-frame images can be confusing, especially for complex scenes (Figure 1a) and lead to ambiguities in their interpretation, indeed similar images are the basis of a number of popular optical illusions. For the trained eye, however, the wire-frame image has its uses. For a start it can be drawn very much more quickly than is the case when the hidden lines are eliminated. The hidden line image is itself ambiguous, though in a different way from the wire-frame, in that you do not know what is round the back (Figure 1b). Many designers, therefore, prefer to work with wire-frames and only incur the expense of producing a hidden line drawing at a late stage, perhaps for client presentation.

The Holy Grail is ever increasing photographic realism of the computer generated image

Another technique which can be used is to display the wire frames as stereoscopic pairs, either in two colours or, on special stereo displays. Looking at these images through an appropriate viewer can give a very convincing impression of 3D. Unfortunately, up to 10% of the population find it difficult to merge the two images successfully.

So far we have talked exclusively about hidden lines. In fact there are two basic

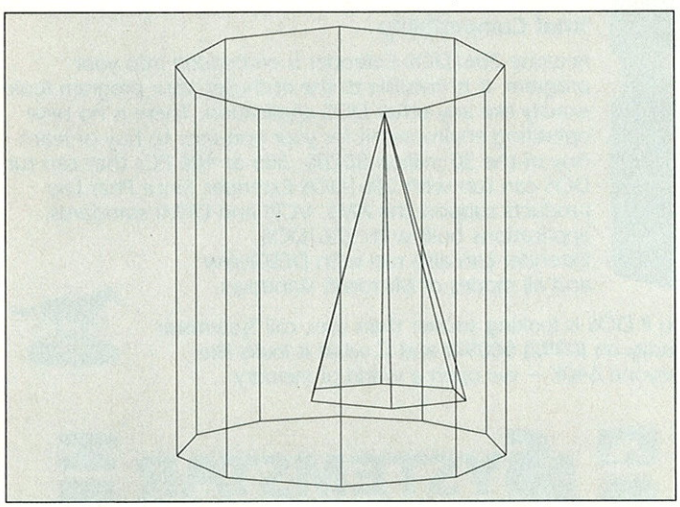


Figure 1a - A confusing wire-frame picture...

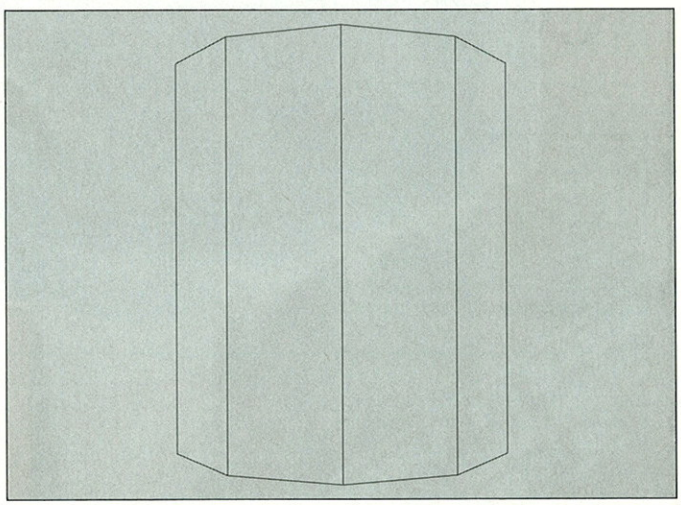


Figure 1b - ... but with hidden lines removed you don't know what's round the back

System Science

• Software • Specialists • Software • Specialist • Software • Specialists • Specialist • Software • Specialists •

Unix

SCO Unix Op. Sys **SCO** Unix Dev Sys.
SCO Xenix 386 **SCO** Xenix 286
SCO Open Destop **SCO** ODT Dev Sys
SCO TCP/IP **SCO** NFS
Interactive Unix Op Sy / Architect Series
Informix 4GL & SQL **RM** Cobol-85
LPI Compilers **C Scape** Screen Lib
LPI C++ (native) for Unix - **NEW**
... many more tools and applications

Mathematics

Derive - The Mathematical Assistant
Mathematica 386/387 or Windows (ver 2.)
Mathematica for MAC/ENH
Mathematica for Workstations (Sun, HP...)
MathCad for Windows (new) GrafTool
What's Best Lindo
Tex Chiwriter Sci WP

Windows Development

(see Microsft and Borland)	
CASE:W Corp.	£695.00
Smalltalk/V Windows	£295.00
Btrieve for Windows	£345.00
Multiscope Windows Debugger	£295.00
WindowsMaker Prof	£795.00
Zinc Interface Lib	£175.00
C++ Views	£225.00
Win++ by Blaise	£150.00

Pascal

(see Borland)	
TopSpeed Pascal SE	£115.00
Object Professional	£110.00
Btree Filer MU	£120.00
Asynch Professional	£89.00
Blaise Power Tools Plus	£89.00
Blaise Turbo Asynch Plus	£115.00

PowerBasic	£89.00
many Basic libraries	
PC Logo	£60.00
LMI UR/Forth	£295.00
Smalltalk V/DOS	£69.00
muLISP 90	£225.00
RM Cobol-85	£call.
MS DOS 5.0, QEMM 386 etc	£call.

Microsoft

Quick C for Win **NEW!**	£99.00
Microsoft C 6.0	£225.00
MS Fortran 5.1 (new ver)	£220.00
MS Windows SDK	£225.00
MS Win SDK & C6.0	£349.00
MS Basic Compiler 7.1	£225.00
MS Cobol 4.0 (DOS & OS/2)	£399.00
MS Macro Assembler 6.0	£75.00
MS Visual Basic	£99.00
MS Quick C 2.5	£60.00
MS Quick C & Assembler	£95.00
MS Quick Basic	£60.00
MS Quick Pascal	£60.00
MS Source Profiler (new)	£49.00

Authorised Languages Dealer

Borland

Borland C++ (Dos & Windows)	£229.00
Turbo Pascal Windows	£115.00
Turbo C++	£59.00
Turbo Pascal 6	£75.00
Turbo Pascal 6 Prof	£150.00
Turbo Debugger and Tools	£99.00
Paradox Engine	£275.00
Paradox 3.5	£445.00

Authorised Languages Dealer

Editors

Professional Editor (SPE)	£195.00
Brief 3.1 (with Mouse)	£195.00
Kedit (Xedit for PC)	£110.00
Epsilon	£175.00

Configuration

PVCS for DOS	£345.00
PVCS 5 User	£1295.00
PVCS Config Builder (PolyMake)	£129.00
MKS RCS 5.1 (with MKS Make)	£155.00

Misc

.RTPatch for DOS, Win & OS/2	£245.00
.RTLink Plus	£295.00
Plink 86 Plus	£295.00
MKS Toolkit (Unix Tools)	£155.00
MKS Lex & Yacc	£159.00
Personal Rexx	£110.00
Ghost (software testing)	£145.00
Flowcharting III	£150.00
C Programmers Toolbox	£155.00

... and many more tools and utilities

C++ Compilers

Zortech C++ DOS & Windows	£225.00
Zortech C++ Dev Ed (incl OS/2)	£360.00
Zortech C++ Sci & Eng	£540.00
Zortech C++ Video (6xVHS)	£275.00
JPI TopSpeed C++ SE	£115.00
JPI TopSpeed C++ PE	£175.00
JPI TopSpeed C++ Pro Dev.	£229.00

C Compilers

see Microsoft	
Watcom C 8.0	£245.00
Watcom C386 8.0 PE	£775.00
Aztec C 86 (with ROM support)	£265.00
JPI TopSpeed C PE	£175.00

C Datafile

CodeBase 4.2	£175.00
CodeBase ++	£175.00
Ctree Plus	£345.00
Btrieve for DOS	£345.00

C Communications

Essential Comms	£195.00
Greenleaf CommLib	£195.00
C Asynch Manager (Blaise)	£115.00

C Screens

CScape with Look & Feel	£365.00
Vermont Views	£325.00
Panel Plus II	£275.00
Vitamin C 4.0	£245.00

C Graphics

Essential Graphics	£245.00
PCX Programmers Toolkit	£165.00
Halo	£195.00
Ingraf	£195.00

... many more libraries

Pharlap 386 DOS Ext SDK	£325.00
Pharlap 286 DOS Ext SDK (new)	£325.00

Fortran Compilers

Lahey F77L	£375.00
Lahey F77L-EM/32 with OS386	£845.00
Salford FTN 77-386	£765.00
Salford FTN77-486	£875.00
RM Fortran with Forte	£545.00
Ingraf	£195.00

... many Fortran Libraries

- **Prices are exclusive of VAT.**
- **Prices include delivery to GB.**
- **Prices are subject to change.**
- **VISA, Access and Mastercard welcome with telephone orders.**

3-5 Cynthia St
London N1-9JF
Fax: 071 837 6411

071 833 1022

• Software • Specialists • Software • Specialist • Software • Specialists •

problems which, though related, are amenable to different solutions. In the days when the only generally available graphics devices were confined to drawing lines (usually dull green on a duller green background), and image hard copy output device was a pen plotter, the challenge was to remove from the image those edges and parts of edges which were invisible from the chosen view point. This is the hidden line problem which, when solved, gives plain lines on a background. Later, as the technologies developed, people wished to have images of greater realism with the surfaces shaded in a naturalistic manner. Multicolour raster graphics is a natural for this kind of work and has lead to the development of techniques which exploit the raster display to do hidden surface removal.

If you remove hidden lines, you don't know what's behind

After the pioneering work at JPL and the University of Utah the techniques of rendering hidden surface images has blossomed, with the Holy Grail being ever increasing photographic realism of the computer generated image. The price has been ever increasing cost of producing a single image, hours of quite powerful computing for one picture being quite the norm.

Today's article goes back a generation or so and looks at one of the classical algorithms, namely that originally proposed by John Warnock in the late 1960's. This algorithm is independent of the display mechanism. Later articles will look at techniques which exploit the peculiar characteristics of raster displays for hidden surface removal and rendering.

```
// Basic 'Warnock' Algorithm
#include <dos.h>
#include <io.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <video256.h>

#define MAXVERTEX 1024
#define MAXPOLY     32
// Maximum screen resolution +1
#define MAXWINSTK 1025
#define BACKGROUND  85
#define BIG          1.0e38

int     BBox(int i);
float   Depth(int n);
void    Display(int colour);
void    Fatal(unsigned char *stg);
void    GetData(void);
int     Nearest(void);
void    PushWindow(int x,int y,
                   int size);
void    PopWindow(int *x,int *y,
                  int *size);
void    SetupGraphics256(void);

int     NumPolygons,
        WindowSP,
        WSize,
        HRes,VRes,HCen,VCen,Type;

float   NumVertex,
        Gamma;

struct  VertexStruct
{ float x,
        y,
        z;
}       Vertex[MAXVERTEX];


struct  PolygonStruct
{ int   firstvertex,
        nvertex,
        colour;
  float a,
        b,
        c,
        d,
        xmin,
        xmax,
        ymin,
        ymax;
}       Polygon[MAXPOLY];


struct  WindowStruct
{ int   x0,
        y0,
        size;
}       WindowStack[MAXWINSTK],
        Window;
//-------------------------------------
void main()
{ int   disjoint,i,pnumber;

  Type=5; HRes=640; Gamma=1.6;
  SetupGraphics256();

  GetData();

  WindowSP=0;
  PushWindow(0,0,WSize);
  while (WindowSP!=0)
  { PopWindow(&Window.x0,&Window.y0,
                         &Window.size);
    // initialise the polygon counter
    i=0;
    disjoint=0;
    while ((i<NumPolygons) &&
                        (disjoint==0))
    { disjoint=BBox(i);
      i++;
    }
    // Something in the window?
    if (disjoint>0)
    { if (Window.size>1)
    // Subdivide if bigger than 1x1 pix
      { Window.size/=2;
       PushWindow(Window.x0+Window.size,
                  Window.y0+Window.size,
                      Window.size);
        PushWindow(Window.x0,
                   Window.y0+Window.size,
                     Window.size);
        PushWindow(
                  Window.x0+Window.size,
                    Window.y0,Window.size);
        PushWindow(Window.x0,
                   Window.y0,Window.size);
      }
      else
      // Find number of nearest polygon
      { pnumber=Nearest();
        if (pnumber>=0)
          Display(
            Polygon[pnumber].colour);
        else        // Window empty
          Display(BACKGROUND);
      }
    }
    else      // Window empty
      Display(BACKGROUND);
  }
  getch();
  EndGraphics256();
}
//-------------------------------------
int     BBox(int i)
// Returns 0 if no part of Polygon[i]
// is in current window, otherwise
// returns 1
{ int   xleft,xright,ybottom,ytop;
  xleft=Window.x0;
  xright=xleft+Window.size-1;
  ybottom=Window.y0;
  ytop=ybottom+Window.size-1;
  if (Polygon[i].xmin>xright)
    return(0);
  if (Polygon[i].xmax<xleft)
    return(0);
  if (Polygon[i].ymin>ytop)
    return(0);
  if (Polygon[i].ymax<ybottom)
    return(0);
  return(1);
}
//-------------------------------------
float   Depth(int n)
// Return distance to polygon n
// calculated at window centre
{ float d;
  int   i,xcen,ycen;
  xcen=Window.x0+Window.size/2;
  ycen=Window.y0+Window.size/2;
  if (Polygon[n].c==0.0)
    for (i=1;i<Polygon[n].nvertex;i++)
     d=max(Vertex[i].z,Vertex[i-1].z);
  else
    d=-(Polygon[n].a*xcen+
        Polygon[n].b*ycen+
        Polygon[n].d)/Polygon[n].c;
  return (d);
}
//-------------------------------------
void    Display(int colour)
// Paint the current window with colour,
// inverting y to make coordinates run
// from bottom left upwards
{ int   i,j;
  for (j=VRes-Window.y0-1;
       j>=VRes-Window.y0-Window.size;
       j--)
    for (i=Window.x0;
         i<(Window.x0+Window.size);
         i++)
      SetPixel256(i,j,colour);
}
//-------------------------------------
void    Fatal(unsigned char *stg)
// Fatal error message
{ int   col,row;
  col=32; row=VRes/2;
  DrawString256("Fatal error",col,row,
                                 7,0);
  DrawString256(stg,col,row+16,255,0);
  DrawString256("Press any key",
                col,row+32,255,0);
  getch();
  EndGraphics256();
  exit(1);
}
//-------------------------------------
```

Figure 2 - An implementation of the Warnock algorithm

RECESSION SPECIAL!

PRE-SALE CONSULTING

TAKE 15% OFF ALL PRICES!

POST SALE SUPPORT

NEW & NOTEWORTHY

Drover ToolBox for Windows v1.2 .....£214
A dynamic link library of 200 functions and seven classes for Windows 3.0 development.
Lahey F77L v5.0 .....£511
New features include allocatable arrays and other important Fortran 90 features.
LANtastic 2Mbps Starter Kit £319
Economic alternative to Novell. Links as few as two PCs and grows upto 100s.
MS CD-ROM Programmer's Library £243
Contains over 70 indexed and cross referenced books and reference manuals, with over 20MB of code that can be included in your programs.
MS Excel v3.0 for Windows £199
Incredible price! Order while supplies last!
MS MS-DOS v5.0 Upgrade £69
Loads into extended memory. Network drivers, TSRs in upper memory. Full screen editor, Undelete, Unformat and more!
MS Visual BASIC £129
Create GUIs in minutes without writing a line of code! Then add functionality with a powerful general-purpose language!
NE-2000 Ethernet Kit £154
16-bit high performance network adapter.
Novell NetWare v2.2 .....£ 556
Combines Netware 2.x, SFT, Advanced and ELS I & ELS II into one product.
Oracle Tools & Database v6.0.....£674
Now includes PL/SQL and SQL*DBA.
Viva 2496if 9600bps FAX/Modem £159
9600bps int. Send/receive fax/modem.
Zortech C++ Developer's Edition v3.0 ...£350
Optimizing C++ Compilers for DOS, Windows & OS/2

ARTIFICIAL INTELLIGENCE
Arity Combination for Windows.....£834
PDC Prolog (Turbo Prolog).....£201

ASSEMBLERS & LINKERS
Blinker.....£201
Dis/Doc Professional.....£188
Microsoft Macro Assembler.....£97
Sourcer.....£105
Spontaneous Assembly.....£160

BASIC
Mach 2 for MS QuickBASIC.....£55
Object Script QuickBASIC.....£105
PowerBASIC.....£71

BUSINESS
ABC Flowcharter.....£177
Adobe Illustrator for Windows.....£271
Baler.....£308
Form Tool Gold.....£55
Grammatik IV.....£54
IBM Current.....£213
IBM Storyboard Live.....£276
Micrografx Charisma.....£324
ORG PLUS Advanced.....£60
Quattro Pro Competitive Upgrade.....£88
Show Partner F/X.....£224
SideKick Plus.....£125
Super Project Plus.....£238
Xerox FormBase /Windows.....£297

C
Aztec C Cross Development Systems.....£376
Borland C++ v2.0.....£295
C Communications Toolkit.....£129
C-terp 386 with Source.....£192
CLEAR+ for C.....£120
Control Pak/Windows.....£383
DESQview API C Library.....£110
Essential Communications.....£194
Greenleaf Comm++.....£149
Greenleaf ViewComm.....£420
Informix C-ISAM for DOS.....£147
Lattice C Development Sys v6.0.....£143
Object Script C.....£349
PC-Lint.....£75
Pro-C with Workbench.....£557
Turbo C TOOLS.....£96
Turbo C++ Second Edition.....£63
WATCOM C /386 Professional.....£904
WindowsMaker.....£596
XVT (Windows/ PM/ Macintosh).....£415
Zortech C++ Database Class Library.....£193
Zortech C++ for Windows & DOS.....£196

CADD
3D Drafting.....£199
AutoSketch.....£143
Design Cad 2D.....£145
Design CAD 3D.....£193
DRAFIX CAD for Windows.....£410
DRAFIX CAD ULTRA.....£237
Generic CADD.....£281

CASE
Case: PM for C++.....£1,713
Case: Windows Corporate Edition.....£855
Case: Windows Standard Edition.....£425
Compeditor II.....£225
EasyCASE Plus v3.0.....£425
Oracle CASE Dictionary.....£1,504

COBOL
COBOL spII.....£415
COBOL/2 Compiler with Toolset.....£1,449
Dialog System.....£805
Level II COBOL.....£398
RM/COBOL-85 Development System.....£756

COMMUNICATIONS
Carbon Copy v6.0.....£108
Crosstalk for Windows 3.0.....£117
LapLink Pro v4.0.....£84
Mirror III.....£81
Procomm Plus.....£64
Reflection 1 for Windows.....£289
Remote2 (Host & Caller).....£105
SmarTerm 240.....£202
WinConnect.....£58

DATABASE & FILE MANAGEMENT
Advanced Revelation.....£705
ALPHA/Four competitive upgrade.....£91
AskSam.....£213
Btrieve DOS 3.1 Network for PC.....£383
Btrieve for Windows.....£383
Clarion Personal Developer.....£43
Clarion Professional Developer.....£444
DataFlex.....£393
dbFast/PLUS.....£166
dbFast/Windows.....£239
FairCom Toolbox Special Edition (1-user).....£448
Gupta SQLWindows.....£910
Informix SQL for DOS.....£521
Oracle Tools & Database v6.0.....£670
Paradox v.3.5.....£488
Paradox SQL Link.....£335
R&R Clipper and FoxBASE+ Module.....£42
R&R Relational Report Writer for dBASE.£160
R&R Report Writer for Oracle SQL.....£192
R&R Worksheet Report Writer.....£96
R:BASE Single User.....£469
SQLWindows for Btrieve.....£1,370
Superbase 4 Windows.....£418
UI Programmer.....£320
XQL Relational Data Manager.....£529
Xtrieve Plus.....£403

DEBUGGERS
ClearView-386/486 Debugger.....£318
Multiscope Debugger for DOS.....£114
Multiscope Debugger for Windows.....£240
Periscope II with Breakout Switch v5.0.....£121
Turbo Debugger & Tools v2.0.....£97

DESKTOP PUBLISHING
Adobe PostScript Cartridge.....£297
Adobe Type Align /Windows.....£59
Adobe Type Manager for Windows.....£60
CatchWord OCR.....£136
ColoRIX VGA Paint.....£97
Facelift for Windows.....£58
Fontware Installation Program.....£56
Fontware Typeface Packages.....£77
Go Script Plus.....£179
Inset + Hijaak.....£160
OmniPage Pro.....£598
OmniProof for DOS.....£99
PageMaker.....£493
Persuasion for Windows.....£303
The Complete Page Scanner w/ OCR.....£579
WordScan 286/386 2 Mb.....£383
Xerox Ventura Publisher Gold v3.0.....£504

DISK/DOS/BACKUP
Back-It.....£74
Copy II PC.....£26
Disk Manager.....£35
Norton Commander.....£94
Norton Utilities.....£110
OPTune.....£43
PC Tools with AntiVirus.....£97
Q-DOS II.....£43
SpeedStor Universal.....£61
SpinRite II.....£58
WonderPlus.....£47
XTree Pro.....£54
XTree Pro Gold.....£89

EDITORS
Brief v3.1.....£176
EDT +.....£213
KEDIT.....£97
Norton Editor.....£47
Sage Professional.....£189
VEDIT PLUS.....£110

FORTRAN
Fortran Toolkit 2.....£116
GRAFMATIC & PLOTMATIC.....£155
RM/FORTRAN Development System.....£345
Spindrift Utility Library (F77L).....£79
ULI Maths Functions Library (F77L).....£188

GRAPHICS
Corel Draw.....£374
Designer for Windows.....£455
Dodot.....£111
Draw Applause.....£263
GRASP with ARTOOLS.....£191
GSS GKS Kernal System for DOS.....£415
GSS Graphic Devlpnt Toolkit for DOS.....£555
HALO.....£227
HALO Professional.....£341
Halo Windows Toolkit.....£338
Harvard Graphics.....£301
HiJaak.....£128
MetaWINDOW.....£222
PCPaintBrush for Windows.....£52
Pizazz Plus.....£85
Windows Draw Plus.....£160

HARDWARE
8087-10 Math Coprocessor.....£157
ASCII Internal 2400 Modem.....£68
ASCII Internal 9600bps V.32 with MNP 5..£428
FastMath 83S87-16 Math coprocesor.....£243
Hardcard IIXL 105Mb, 9ms Hard Disk.....£555
Logitech TrackMan Serial Mouse.....£76
MouseMan Serial Mouse.....£68
Pacific Page PS Cartridge for HP IIP.....£348
Paradise 1024x768 VGA w/ 512k.....£191
Practical 2400 Pocket Modem.....£106
ProScan PS1000 64 Grayscale Scanner.....£520
SatisFAXtion Classic.....£375
ScanMan Plus PC.....£171
The Complete Communicator.....£412
The Complete Page Scanner.....£530
VIVA 9600 V42 bis External Modem.....£446
Weitek 3167/25MHz Math Coprocessor.....£558
Worldport 2496 Portable Fax.....£465

INTEGRATED
Framework III.....£424
Lotus Symphony.....£485
Lotus Works.....£89
Microsoft PC Works.....£92

LAN
LANtastic AE-2 Ethernet Starter Kit.....£450
NE-1000 Ethernet Kit.....£125
Netware v. 3.11 20 user.....£2,327
Netware C Interface for DOS.....£196
Oracle for Server.....£1,254
Procomm Plus for Network.....£325
Q-DOS II for Network (8-user).....£134
Saber File Manager.....£200
Saber LAN Administration Pack.....£351
Saber Menu for MS Windows 3.0.....£249
Saber Menu System Single-User.....£45
Saber Meter Usage Tracking Software.....£99
Saber Secure.....£75
SoftwareBridge LAN.....£143
Super Project Expert Lanpak.....£420
Superbase 4 Windows Network.....£598

MAC
Adobe Illustrator Mac.....£332
Adobe Type Manager.....£56
Carbon Copy Mac.....£108
LapLink Mac III.....£84
MathCad.....£305
Mathematica.....£425
Norton Utilities for Mac.....£82
Oracle Developers Kit Mac.....£501
Pagemaker.....£486
PC-Matlab /Mac.....£752
Persuasion.....£311
Quark Xpress.....£512
Smalltalk/V for Mac.....£107
Zortech C++ for MAC.....£196

MEMORY MANAGEMENT
386 Max.....£60
Above Disk.....£58
Headroom.....£70
QEMM 386.....£57
Turbo EMS.....£48
Vcache.....£45

MODULA-2
EmsStorage with Full Source.....£48
Logitech Modula-2 Development System..£158
Logitech Modula-2 Window Pack.....£30
ModBase.....£21

OBJECT ORIENTED PROGRAMMING
C++ Views.....£319
Codebase ++.....£222
HCR/C++.....£727
Smalltalk/V.....£69
Smalltalk/V 286.....£107
Smalltalk/V Communications.....£27
Smalltalk/V EGA/VGA Colour Pack.....£27
Smalltalk/V Goodies 1.....£27
Smalltalk/V-Windows.....£268
Zinc Interface Library.....£160
Zortech C++ MS-DOS 386.....£641
Zortech Windows Toolkit.....£45

OPERATING ENVIRONMENTS
386/ix Operating System Release 3.2.....£330
DESQview.....£77
DR DOS.....£130
DR Multiuser DOS.....£448
GeoWorks Ensemble.....£105
Microsoft Windows.....£92
OS/2 Extended Operating System.....£662
OS/2 Standard Operating System.....£258
SCO UNIX System V/386 v.3.2.2.....£370

OS/2
Corel Draw for OS/2.....£320
Describe Word Publisher (Client).....£164
Display Write 5/2.....£234
Halo for OS/2.....£338
Microsoft OS/2 Presentation Mgr.Softset.....£92
Microsoft OS/2 Presentation Mgr.Toolkit...£308
Oracle OS/2.....£652
PageMaker for OS/2.....£486
Paradox OS/2.....£462
Qedit for OS/2.....£48
QEMM Expanded Memory Manager 50/60.£58
Smalltalk/V-PM.....£268
Super Project Expert/2.....£481
ToolBook (OS/2).....£280
TopSpeed C++ for OS/2 Standard.....£213
Vitamin C for OS/2.....£319
Wide Angle for OS/2.....£62

OTHER
ACT! (Sales lead tracking).....£233
Adlib Card (Sound).....£104
Builder (Batch Compiler).....£81
graphics-MENU (GUI).....£134
Guide 3 for PC (HyperText).....£358
HP NewWave (GUI).....£105
HyperPAD.....£82
Install (Automatic installation).....£148
KnowledgePro.....£420
Menuet (GUI Development).....£209
PolyAWK v1.3.....£126
RPG II Development System.....£1,117
SCHEMA-PCB (SC-4).....£733
SoundBlaster.....£147
System Sleuth (Diagnostics).....£72
VirusCure Plus.....£64

PASCAL
Asynch Plus.....£121
B-Tree Filer.....£97
Graf/Drive Plus Personal.....£81
Object Professional.....£132
Pascal TOOLS.....£81
POWER TOOLS PLUS.....£96
Topaz.....£64
Turbo Analyst.....£90
Turbo Pascal for Windows.....£150
Turbo Pascal Professional.....£190
Turbo Professional.....£90

SCIENCE & ENGINEERING
ChiWriter Chemistry Support.....£45
Derive v2.0.....£183
InGraf.....£263
Labtech Notebook.....£801
Matfor - 286.....£225
MathCad for Windows.....£303
Mathematica 386.....£597
Mathematica for Windows.....£855
PlotLib.....£188
SSP/PC Scientific Subroutine Library.....£245
Tech*Graph*Pad.....£335
Zortech C++ Science & Engineering.....£490

SCREEN DESIGN & MANAGEMENT
C Windows Toolkit.....£86
Flash-Up.....£69
Hi Screen XL.....£112
POWER SCREEN.....£96
ProPak.....£412
SayWhat!?.....£33
Vermont Views.....£319
Vitamin C.....£254
WNDX GUI Development System.....£321

TRANSLATORS
BAS_C Commercial.....£242
FOR_C ++.....£870
FOR_C with Runtime Source.....£628
Pascal to C Translator.....£140
Turbo Plus.....£138

UNIX/XENIX
Architech Series Workstation Developer..£1,528
DataFlex for UNIX 386.....£950
Informix C-ISAM (SCO 4.0).....£295
LPI COBOL Compiler.....£1,076
Norton Utilities for System V v3.2.....£174
Oracle UNIX/Xenix 386.....£1,476
SCO UNIX System V/386 Dev.System.....£641
VEDIT PLUS for UNIX/XENIX.....£184

WINDOWS DEVELOPMENT
Asymetrix ToolBook.....£280
BeckerTools.....£73
Microsoft C with Windows SDK.....£466
MS Windows Device Driver Dvlpnt Kit.....£311
Microsoft Windows Productivity Pack.....£38
Object Vision.....£86
WIN++.....£160
WindowsMaker Professional.....£746
WinSleuth.....£72

We will match any advertised Price!

- Prices exclusive of V.A.T.
- All major credit cards accepted.
- 30 day invoicing for qualified organizations.
- Special discounts for large or corporate buyers.
- Prices subject to change without notice.

UK8/91

USA SOFTWARE

2 OLD BROMPTON ROAD • SUITE 315 • LONDON SW7 3DQ
PHONE: 071–351 2360 • FAX: 071–352 0974

Over 3,500 products carried. Call today for price list-disk.

CIRCLE NO. 208

Clipper Heads...

Let Me Tell You About

PARADOX®

The Babelfish Data Driver for Clipper is Finally here.

Our Paradox Data Driver unleashes the power of Clipper by giving you easy access to the multi-user freedom of the Paradox database environment.

The Babelfish Paradox Data Driver gives the Clipper developer complete compatibility with:

* Paradox data files
* Paradox primary & secondary indexes
* Multi-user network handling files
* Clipper Summer '87 & 5.0x versions

QBS Software Ltd

10 Barley Mow Passage, Chiswick,
London W4 4PH Tel: 081 994 4842
Fax: 081 994 3441 BBS: 081 747 1979

30 days money back guarantee.
90 days technical support
Just £349.00 plus delivery plus VAT

CALL NOW FOR FREE DEMO DISK

SuccessWare℠

All Trademarks Recognised

CIRCLE NO. 209

Back Planes

It is important to simplify the removal problem as much as possible in order to speed the process. One simple initial step is to remove all backward facing planes from further consideration. The way to do this is to construct a database for objects in which the vertices around faces are numbered scrupulously either all clockwise or all anti-clockwise (such objects are said to be *orientated*). Suppose an anti-clockwise ordering is chosen. Once all the vertices of an object have been projected into the 2D screen coördinates, the vertices of the visible faces will be wrapped in an anti-clockwise direction while the hidden faces will be clockwise. The direction of wrapping is tested by taking the *vector product* (alternatively referred to as the *cross product*) of two adjacent edges. If the vector product is positive, the labelling of the vertices is anti-clockwise and so the face is retained. If the product is negative the ordering is clockwise so you have a back face. If the product is zero the two edges are collinear, so try again with another pair.

There is no unique Warnock algorithm

To calculate the vector product suppose that the coördinates of the end vertices of two adjacent edges are:

$(x_0,y_0), (x_1,y_1), (x_2,y_2)$

The vector product is:

$(x_1-x_0)*(y_2-y_1)-(x_2-x_1)*(y_1-y_0)$

A similar calculation can be used to determine whether a point lies inside a closed polygon. Suppose that the coördinates of the point are (x,y) and the polygon is orientated anti-clockwise with vertex coördinates $(x_0,y_0), (x_1,y_1) \ldots . (x_n,y_n)$. Working round the sides of the polygon we calculate

$(x-x_0)*(y_1-y_0)-(y-y_0)*(x_1-x_0)$

$(x-x_1)*(y_2-y_1)-(y-y_1)*(x_2-x_1)$

etc and for the last side

$(x-x_n)*(y_0-y_n)-(y-y_n)*(x_0-x_n)$

If the result of each calculation is negative the point (x,y) lies inside the polygon. Should any result be positive the point is outside. A result of zero means that the point may be exactly on an edge of the

```
void    GetData(void)
// Forced test data
{ WSize=128;
  NumVertex=14;
  Vertex[0].x= 90.0; Vertex[0].y= 10.0;
  Vertex[0].z= 80.0; Vertex[1].x= 90.0;
  Vertex[1].y=110.0; Vertex[1].z= 80.0;
  Vertex[2].x= 80.0; Vertex[2].y=110.0;
  Vertex[2].z= 80.0; Vertex[3].x= 40.0;
  Vertex[3].y= 10.0; Vertex[3].z= 80.0;
  Vertex[4].x=110.0; Vertex[4].y= 50.0;
  Vertex[4].z= 40.0; Vertex[5].x=110.0;
  Vertex[5].y= 80.0; Vertex[5].z= 40.0;
  Vertex[6].x= 20.0; Vertex[6].y= 80.0;
  Vertex[6].z= 40.0; Vertex[7].x= 20.0;
  Vertex[7].y= 50.0; Vertex[7].z= 40.0;
  Vertex[8].x=120.0; Vertex[8].y= 40.0;
  Vertex[8].z= 20.0; Vertex[9].x=100.0;
  Vertex[9].y=100.0; Vertex[9].z= 20.0;
  Vertex[10].x= 60.0; Vertex[10].y= 60.0;
  Vertex[10].z=120.0; Vertex[11].x= 20.0;
  Vertex[11].y= 20.0; Vertex[11].z= 100.0;
  Vertex[12].x= 90.0; Vertex[12].y= 20.0;
  Vertex[12].z= 100.0; Vertex[13].x= 55.0;
  Vertex[13].y= 90.0; Vertex[13].z= 100.0;
  NumPolygons=4;
  Polygon[0].firstvertex=0;
  Polygon[0].nvertex=4;
  Polygon[0].colour=7;
  Polygon[0].a=0.0; Polygon[0].b=0.0;
  Polygon[0].c=1.0; Polygon[0].d=-80.0;
  Polygon[0].xmin=40.0;
  Polygon[0].xmax=90.0;
  Polygon[0].ymin=10.0;
  Polygon[0].ymax=110.0;
  Polygon[1].firstvertex=4;
  Polygon[1].nvertex=4;
  Polygon[1].colour=56;
  Polygon[1].a=0.0; Polygon[1].b=0.0;
  Polygon[1].c=1.0; Polygon[1].d=-40.0;
  Polygon[1].xmin=20.0;
  Polygon[1].xmax=110.0;
  Polygon[1].ymin=50.0;
  Polygon[1].ymax=80.0;
  Polygon[2].firstvertex=8;
  Polygon[2].nvertex=3;
  Polygon[2].colour=63;
  Polygon[2].a=1.875;
  Polygon[2].b=0.625;
Polygon[2].c=1.0;
Polygon[2].d=-270.0;
  Polygon[2].xmin=60.0;
  Polygon[2].xmax=120.0;
  Polygon[2].ymin=40.0;
  Polygon[2].ymax=100.0;
  Polygon[3].firstvertex=11;
  Polygon[3].nvertex=3;
  Polygon[3].colour=255;
  Polygon[3].a=0.0; Polygon[3].b=0.0;
  Polygon[3].c=1.0; Polygon[3].d=-100.0;
  Polygon[3].xmin=20.0;
  Polygon[3].xmax=90.0;
  Polygon[3].ymin=20.0;
  Polygon[3].ymax=90.0;
}
//-------------------------------------
int     Nearest(void)
// Returns the index to the nearest
// polygon, calculated at the centre of
// the 1x1 pixel window
// Returns -1 if the window is empty
// For each polygon, test the pixel
// centre against each edge taken
// anti-clockwise
// The point is inside the polygon if
// the test product is >0 for all edges
{ float z,zmin;
  int   i,j,n,p1x,p1y,p2x,p2y,pinside,
        px,py,pz,vx;
  n=-1; zmin=BIG;
  // Pixel centre
  px=Window.x0+Window.size/2;
  py=Window.y0+Window.size/2;
  for (i=0;i<NumPolygons;i++)
  { vx=Polygon[i].firstvertex;
    for (j=0;j<Polygon[i].nvertex-1;j++)
    { p1x=Vertex[vx].x;
      p1y=Vertex[vx].y;
      p2x=Vertex[vx+1].x;
      p2y=Vertex[vx+1].y;
      pinside=(px-p1x)*(p2y-p1y)-
                (py-p1y)*(p2x-p1x);
// Pixel centre lies outside the polygon
// for jth edge
      if (pinside>0) goto skip;
      vx++;
    }
// Deal with last edge
    p1x=Vertex[vx].x; p1y=Vertex[vx].y;
   p2x=Vertex[Polygon[i].firstvertex].x;
 p2y=Vertex[Polygon[i].firstvertex].y;
    pinside=(px-p1x)*(p2y-p1y)-
                (py-p1y)*(p2x-p1x);
// Pixel centre inside the polygon for
// all edges
    if (pinside<=0)
    { z=Depth(i);
      if (z<zmin)
      { zmin=z; n=i;
      }
    }
    skip:;
  }
  return (n);
}
//-------------------------------------
void    PushWindow(int x,int y,int s)
// Push a window on to the stack
{ WindowStack[WindowSP].x0=x;
  WindowStack[WindowSP].y0=y;
  WindowStack[WindowSP].size=s;
  WindowSP++;
  if (WindowSP>=MAXWINSTK)
       Fatal("Window stack overflow");
}
//-------------------------------------
void    PopWindow(int *x,int *y,int *s)
// Pop a window off the stack
{ WindowSP--;
  if (WindowSP<0)
       Fatal("Window stack underflow");
  *x=WindowStack[WindowSP].x0;
  *y=WindowStack[WindowSP].y0;
  *s=WindowStack[WindowSP].size;
}
//-------------------------------------
void    SetupGraphics256(void)
{ int   i;
  union REGS regs;
  unsigned char stg[8],text[81];
  VRes=(3*HRes)>>2;
  HCen=HRes>>1; VCen=VRes>>1;
 // Deal with some 'funny' BIOSs
 // like the Ti'Ko one
  if (Type==5)
  { switch (HRes)
    { case 640: regs.x.ax=0x2e; break;
      case 800: regs.x.ax=0x30; break;
      default:  regs.x.ax=0x38;
    }
    int86(0x10,&regs,&regs);
  }
  InitGraphics256(Type,HRes);
  SetDefaultPalette256(Gamma);
}
//-------------------------------------
```

Figure 2 - An implementation of the Warnock algorithm (continued)

polygon; *may*, because in reality the test is against lines of infinite extent. If you want to know that the point is really on an edge and not some projection of it, it would be necessary to carry out further tests.

The Warnock Algorithm

The basic idea underlying the Warnock algorithm is very general and is rooted in a model of how we perceive and analyse a scene which we are viewing. The idea is to expend as small an amount of effort as possible on areas which contain little or no information but to concentrate instead on those where the information content is high.

The algorithm deals with what is displayed, ie it works in the image space after any viewing transformations have been carried out. It first considers the viewing window to determine if it is empty or if it contains something which is simple enough to display. If not the window is subdivided and the process repeated for each sub-window.

Ultimately sub-windows become as small as the desired limit of resolution. The remaining information is then evaluated and displayed in a single colour. If desired anti-aliasing of the image can be accomplished by continuing the subdivision process to a resolution which is finer than that of the display system and then averaging the sub-pixels to determine the displayed pixel's attributes.

Individual implementations of this idea differ as to what is regarded as 'too complicated to draw, so subdivide'. The details of the subdivision, too, are amenable to much refinement. The simplest idea is just to chop the window into four equal pieces. More sophisticated implementations search for polygons and adjust sub-windows to fit round them in the hope of wasting less time on parts of the scene which turn out to be background. This method shows significant improvements of performance on simple scenes. For complicated scenes their advantages are less and the simple approach is probably just as good.

It is easy to see, then, that there is no unique Warnock algorithm. Indeed this versatile method can be used for hidden line or hidden surface removal simply by changing what is done when a pixel-size window is reached. For hidden lines, if an edge does not pass through the pixel it is given the background colour. For hidden surfaces the nearest polygon, if any, is determined and the colour of that polygon, which may be patterned or shaded, given to the pixel.

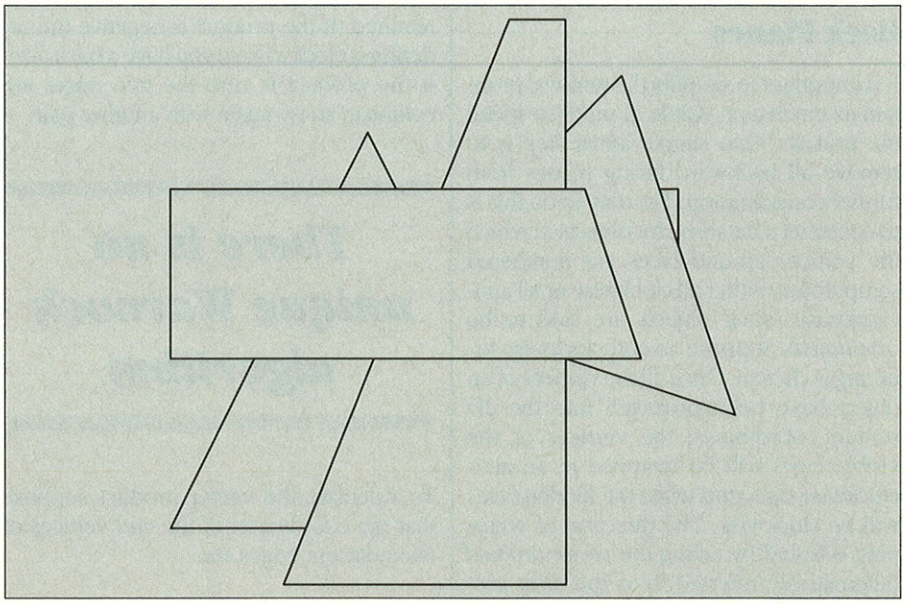


Figure 3 - Line drawing of program output

The Code

In this article we will consider one of the simplest possible implementations, stripped of complications which tend to obscure what is going on. A corollary is that the number of subdivisions of the window and the depth of stacking is maximised! If so inspired, readers can use it as the basis of more complex variants of their own.

The Warnock algorithm is extremely versatile and can be used for both hidden line and hidden surface removal

The program uses the super-VGA graphics library previously published in .EXE Magazine. The values of the variables `Type` and `HRes` should be adjusted to suit your own particular display. The code is set-up for hidden surface removal. If the window size is greater than one pixel and contains anything of interest the algorithm always subdivides. For windows greater than pixel-size, a simple bounding box test is used to find if any part of a polygon lies within the window. For pixel-size windows, the visible polygon, if any, is determined by first finding the depth of all polygons at the centre of the pixel.

No depth priority sorting is carried out, nor is advantage taken of prior knowledge about window-polygon relationships. The coördinates in the array `Vertex` are assumed to have undergone all the necessary viewing transformations.

With the given data the program should output a solid shaded image, a line drawing of which is shown in Figure 3.

EXE

Dr Graeme Webster was formerly Head of Department of Computer Science and later Deputy Director, Academic, of Teesside Polytechnic. He has been involved with computer graphics for the last 20 years with an especial interest in 3D visualisation for Designers. Currently setting up a Centre for Scientific Visualisation under the aegis of the Teesside Development Corporation and Senior Research Fellow at Teesside Polytechnic.

As usual, the code given with this article, together with all the other Webster programs featured in earlier issues, is available on disk. Send a blank floppy disk to the Editor, following exactly the instructions on Page 1. Mark your envelopes '3D-GRAPHICS'.

Coming soon... Dr W has promised us a cracker for our December issue.

C Programmers please note

Scripton© + C = PostScript®

Scripton is a library of functions that allows programmers to produce PostScript code from C. PostScript is an interpreted language that describes page layouts, allowing unrestricted control of PostScript printers, including Laser image setting machines. Graphics and type may be combined with overprinting, stretching and shrinking, grey levels, rotation and all the fonts that your printer provides. This advertisement was made using a C program, Scripton and a Laser Imagesetter.

Scripton features

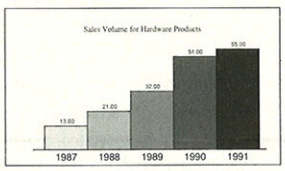
Typographic control `Typographic control` Typographic control

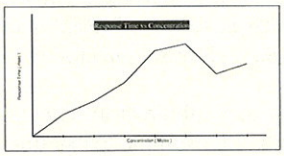
Printing in any direction can be accomplished easily.

Scripton

Scripton

Graphics Graphics Graphics





For a FREE Demo disk contact QBS.

Scripton costs £155.00 + VAT for Disk, Manual & Support

QBS Software Ltd, 10 Barley Mow Passage, London W4 4PH
Tel : 081-994-4842 Fax: 081-994-3441

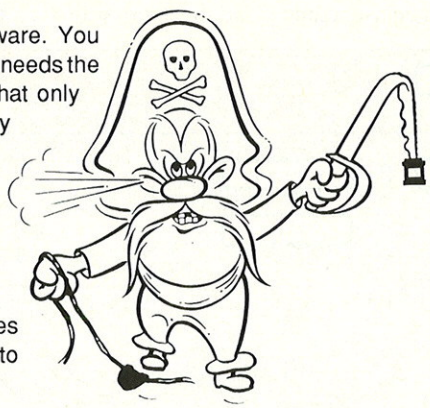
PostScript is the trademark of Adobe Corporation

CIRCLE NO. 210

BRENT Communications and Telemetry Consultants

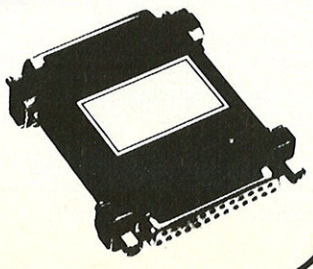
BEWARE THE PIRATE'S PATCH

You sell your software. You don't give it away. It needs the kind of protection that only a top quality UN-PATCHABLE dongle affords, but you don't want to pay the Earth for it, and you want to be sure that you'll not be making mistakes in incorporating it into your code.



The MAXPRO system is for you. There are microprocessor based units at realisitc prices which take care of complete .EXE files without access to source code. Set stop dates, tamper detection and many other facilities on a menu-driven front end. Encrypt in just moments. MAXPRO even copes with such as Clipper, QB & Clarion files with internal overlays. Neat trick.

For additional information contact us at



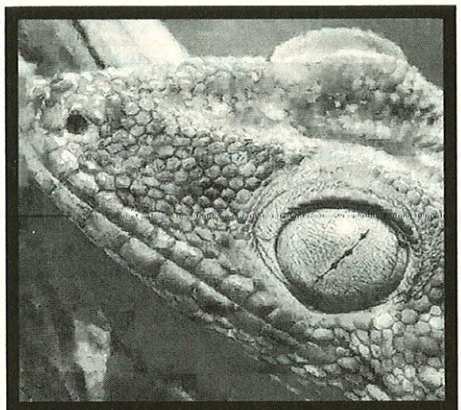
Brent Communications
Unit 2
Dragon Industrial Estate
Harrogate HG1 5DN
Tel: (0423) 566972
Fax: (0423) 501442

CIRCLE NO. 211

✱100 Full Colour Images
✱100 Pages of Text
✱ONE Floppy Disk

Image decompression and display in SOFTWARE ALONE.

See for yourself the fantastic space savings achieved using Fractal Image Compression. Buy the world's first Floppy Book containing 100 full colour images together with 100 pages of text, index, interactive browsing software and software decompression – all on a single floppy disk.



When you've seen images such as the Gecko above compressed from an original file size of 768Kb to just 10Kb and then decompressed and displayed on your own PC in real time and in software alone you'll be amazed and itching to build Fractal Image Compression into your own applications or to publish your own Floppy Books.

FLOPPY BOOK™

Fractal Image Compression, based on the Fractal Transform™ invented by Dr Michael Barnsley, co-founder of Iterated Systems, is set to revolutionise the world of image handling by enabling images to be stored simply and economically and decompressed without the need for expensive additional hardware. This P.OEM™ book provides a model for users of this new technology, allowing storage of a photographic, 24-bit, true-colour image in only 10Kb of disk space. Iterated Systems develops and markets a range of Software Developers Systems to enable the high level developer to incorporate Fractal Image Compression and Decompression into their applications.

System requirements are MS or PC-DOS version 3.00 or higher, Intel 80286 processor or higher, VGA graphics and a 3.5" 1.44Mb floppy drive. Software is supplied under licence, a copy of which is available on request.

Yes, I want to learn more about Fractal Image Compression

Please send me _____ copies of P.OEM™ Floppy Book

I enclose a cheque for _____ (£20.00 per copy) made payable to Iterated Systems Ltd.

☐ Please send me more information on Iterated Systems and their products

Name _____

Job Title _____

Company _____

Address _____

_____

Phone _____

Send to:

Iterated Systems Ltd

Wyvols Court, Swallowfield,
Near Reading, Berkshire, RG7 1PY
Phone: (0734) 880261 Fax: (0734) 880360

P.OEM, Fractal Transform and the 'Fern' logo are Trademarks of Iterated Systems Ltd. Floppy Book is produced by Jones and Bartlett in cooperation with Iterated Systems.

CIRCLE NO. 242

Windows Graphics Without Tears

Cliff Saran takes a look at an object-oriented graphics library and a presentation graphics tool for Windows.

Take one look at the Windows GDI and you will certainly feel that there must be an easier, more intuitive way to produce graphics for Windows applications. The first hurdle to overcome is Windows' concept of a Display Context (DC). Everytime an application needs to draw something in a window, it must obtain a handle to this DC. There are almost 200 functions in the Windows GDI; some of these require a DC handle and all take different numbers of parameters. A possible solution is the addition of an interface library that sits on top of the GDI in such a way that an applications programmer would be insulated from the GDI itself. Graphics are inherently object-oriented (eg a given shape on the display screen will have a number of attributes associated with it) so it would be even better if this library were object-oriented. ObjectGraphics from Whitewater and ChartBuilder Control from Bits Per Second are two such tools for the aspiring Windows graphics programmer and, in this article, I hope to illustrate some of their main features.

ObjectGraphics

The ObjectGraphics Library (OGL) works as an extension of the ObjectWindows class library, providing an object-oriented interface to the Windows GDI. It is currently available for Turbo Pascal and Actor, although a Borland C++ version should be available shortly. The review copy was the Turbo Pascal For Windows version. It is supplied with a single user manual and the software, which is provided on both 5 1/4" and 3 1/2" disk formats, may be installed by running the `Setup` program from the Windows Program Manager. Once installed, OGL occupies almost 700 KB of disk space and it creates four subdirectories.

The `Demos` subdirectory contains source code for 15 example programs. OGL is shipped with an incredible graphical editor called ObjectDraw which was itself written using the ObjectGraphics library. This program lets you draw shapes using rubberbanding to stretch the shape (it even works with ellipses). It also provides controls for zooming in and out of an image. Diagrams may be saved as metafiles or as bitmaps and hardcopies can also be produced. The entire source code for the ObjectDraw appli-

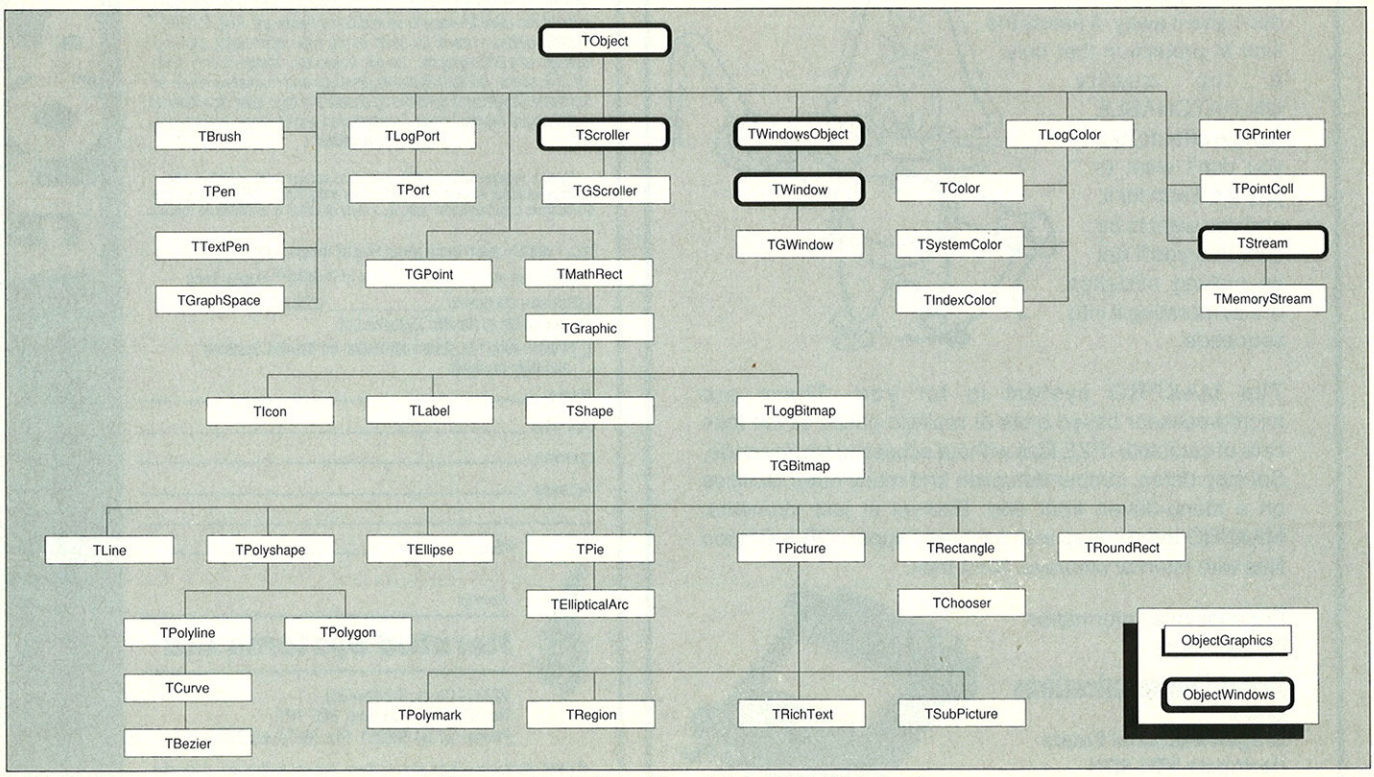
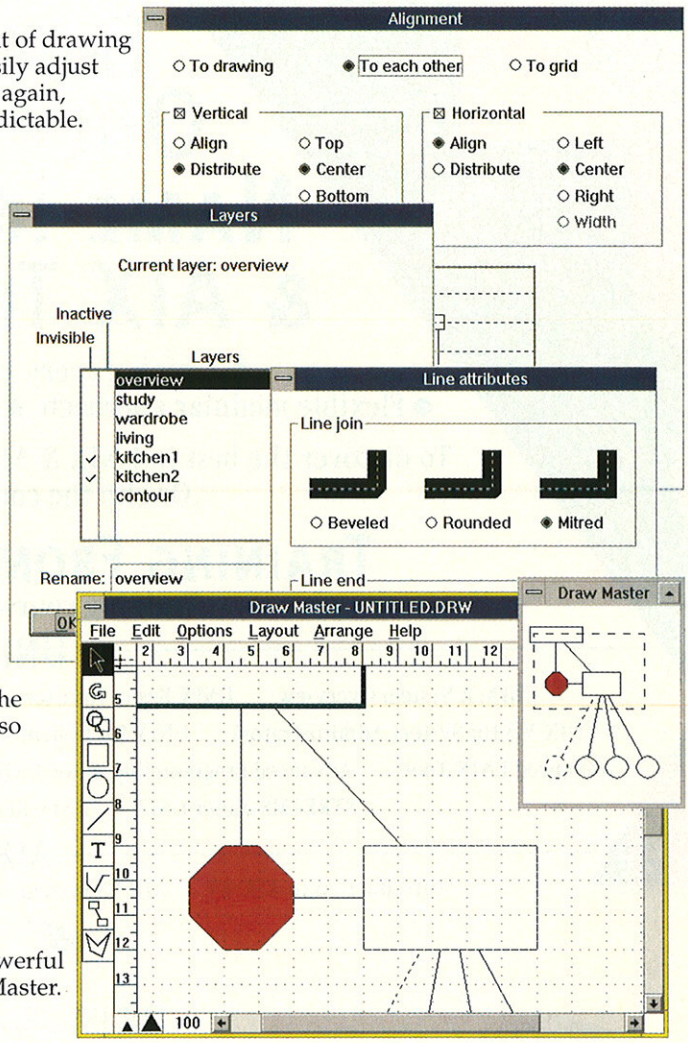


Figure 1 - ObjectGraphics class hierarchy

Will DRAW_Master save you a lot of time and money too?

DRAW_Master is a construction kit for the easy development of drawing tools. If you need to build a drawing application you can easily adjust DRAW_Master to your exact needs and use it over and over again, making the development of drawing tools easy, fast and predictable.



Product of the _Master Series

DRAW_Master was built as an extension to our GUI_Master object-oriented development environment for OS/2 PM and Microsoft Windows. GUI_Master itself consists of an Interface Builder, a Class Browser and a class tree with 85 user interface classes.

High Functionality

DRAW_Master consists of a class tree of more than 120 classes. Together they form a drawing application which offers a lot of standard functionality such as undo/redo, copy/paste, zoom in/zoom out, alignment, patterns, drawing size. In addition it has features like layer control, overview window and resizing. You can use DRAW_Master as the starting point for your application. By subclassing you can adjust and expand your new application. With DRAW_Master you can build a new drawing tool in a few weeks.

Source code included

To give you the best benefit of DRAW_Master we included the source code. A cookbook and a class reference manual are also included.

Prerequisites

- C++ 2.x compiler
- SDK (for OS/2or MS Windows)
- GUI_Master (source license advised).

Free drawing tool included

If you order now you will receive our OS2DRAW tool. A powerful drawing tool in the OS/2 environment made with DRAW_Master.

Prices

DRAW_Master source code license	$15,000.=
GUI_Master source code license	$50,000.=

There are no runtime licenses

Request for Information

VSR Holland
P.O. Box 2584
3500 GN Utrecht
The Netherlands

Fax: Intl. +31 30 31 04 26
Phone: Intl. +31 30 32 49 44

Name _____

Company name _____

Department _____

Street _____

City _____ State _____ Zip _____

Country _____ DRAW_A4_0991_C

FEATURES

- Patterns
- Object selection
- Printing
- Multiple undo/ redo
- Changeable palette
- Rotation
- Zoom in/ Zoom out
- Polygon
- Connections
- Scrolling
- Auto scrolling
- Rulers
- Grid
- Reshaping
- Fonts
- Background/ foreground
- Grouping/ ungrouping
- Automatic Page numbering
- Page breaks
- Polylines
- Layering
- Colouring
- Keyboard shortcuts
- Resizing
- Copy/paste
- Duplication
- Dashed Lines
- Line width
- Line joins
- Line ends
- Grid snapping
- Overview
- Alignment
- 135 changeable classes
- Reuse of code
- Adding new classes
- Class reference
- Cookbook

CIRCLE NO. 214

cation can be found in the `Draw` subdirectory. Finally there is the library itself and this is divided into three TPU files (OGL1, OGL2 and OGL3) in the `Units` directory.

Unlike Borland's ObjectWindows, the manual that's provided with OGL is rather sparse (about 236 pages in total). It includes some useful examples on programming with OGL, but there is no reference section. I found that in order to write even the simplest of programs, I spent considerable time browsing through the on-line help, searching for clues of how to use an objects methods. Even the example programs appear to be over-complicated. I was assured, by Whitewater, that a future release of the product would contain more simpler, example programs.

OGL contains a number of pre-defined shape classes including Rectangles, Ellipses, Arc and Polygons. Each shape owns a set of properties eg its colour or the width of its outline, and these attributes are also given their own object types. In addition, OGL provides a powerful container class for building complex pictures that are constructed as a linked-list of shapes. Figure 1 illustrates the OGL class hierarchy.

Producing Graphical Output

In graphics, a shape is normally drawn onto a physical device. OGL provides three actual output device classes (`TGWindow`, `TGPrinter`, `TGBitmap`) for this purpose. These all share a common protocol and each defines a `Port` and a `Space` field. The `Port` field holds a `TPort` object which provides the developer with a virtual output device. In order to draw an object onto a physical device, the device must first be *Associated* with this Port. The drawing process is then performed and finally the device must be *Dissociated* from the Port.

The `Space` field points to a `TGraphSpace` object. This provides the mappings for the coördinate transformation mechanism that is invoked when rendering a drawing onto a physical device. OGL permits an application to choose its own system of coördinates to model a given situation in the real world. The transformation is realised by initialising three rectangular areas to represent the *World Rectangle*, the *Mapping Rectangle* and the *Display Rectangle* respectively. The World Rectangle defines the upper and lower boundaries of an arbitrary coördinate system that is used to model real-world measurements. Similarly, the Mapping Rectangle defines a rectangular area in this 'world' that will appear on the display. The Display Rectangle contains the actual device coördinates that determine the size and position of the Viewport onto which the Mapping Rectangle will be displayed. OGL permits both the World and the Mapping rectangles to be changed, although it is not possible to alter the settings of the Display rectangle since these are fixed by the device itself.

OGL also provides methods that respond to movement of the mouse or scroll-bar by returning their positions in world coördinates, so that a developer need never worry about actual device coördinates. In the case of the scroll-bar, a new `TGScroller` class is used to handle these events.

Rendering Tools

In OGL, a rendering tool is considered to be a device that allows a number of drawing attributes to be bundled into a single structure. There are four rendering tools: the first three providing the familiar pen, brush and text pen attributes, and the last controls the coördinate system of the application using the *GraphSpace* attributes covered above. These rendering tools all provide an `Init` constructor, an `InitDefault` constructor and a `Build` method. The `InitDefault` constructor is a simplified version of `Init` that sets all attributes to their default values while the `Build` method may be used to change all attributes at a later stage, after the rendering tool has been initialised.

```
procedure TDrawWindow.BeginDrag
      (MousePt :PGPoint; KeyStates :Word);
      .
      .
      .
{ Fetch pointer to object selected with mouse.}
GObj := Picture^.ThatContains(MousePt);
if GObj^.Id = X_AxisID then
begin
  {Create a new Textpen for the label}
  LabelName := New(Plabel,
    Init(Xc, Yc, LabelText, tc_Tools));
  Port^.Associate(@Self);
  { Write label to port using }
  {Associate/Disspociate protocol.}
  LabelName^.Draw(Port);
  Port^.Dissociate;
  Dispose(LabelName, Done);
end;
if GObj^.Id = Y_AxisID then
begin
 { Write Label as above.}
end;
if GObj^.Id = (BarId+0) then
begin
  {Bar 0 has been clicked so draw}
  { dialog box with more info.}
  MessageBox(HWindow,
    Mess1, Title1, mb_OK);
end;
{
  Since each bar has a unique ID, the
  other bars can be selected by simply
  writing code for each ID.
}
```

Figure 2 - Code for Mouse Event Handler

Pens control the appearance of lines using a set of five attributes. The *Line Style*, *Color* and *Line Width* attributes determine the obvious properties of a line (eg a solid blue line, four pixels wide). The *Combination* control is used to set the logical drawing operation that will be performed when the line is drawn. There are eight in total including the *And*, *Or*, *Xor*, *Set* and *Inverse* operations which all control how the line colour is combined with the background. Finally, when drawing a segmented line, the *Transparency* attribute controls how the gaps between lines will appear (ie set to background colour - opaque, or transparent). All of the shapes in the OGL library are drawn with lines, thus their outlines may be changed using this rendering tool.

Unlike Pens, Brushes only have two attributes (Colour and Pattern) which control how the interior of a closed shape will appear. There are 19 built-in bitmap patterns in the OGL library and up to 10 user-defined bitmap patterns can be added.

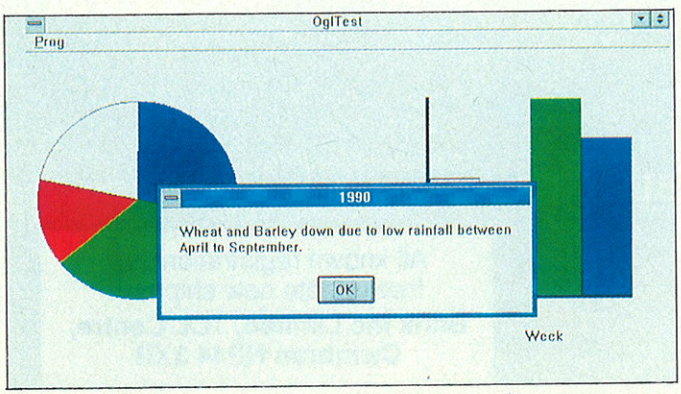


Figure 3 - An ObjectGraphics Application

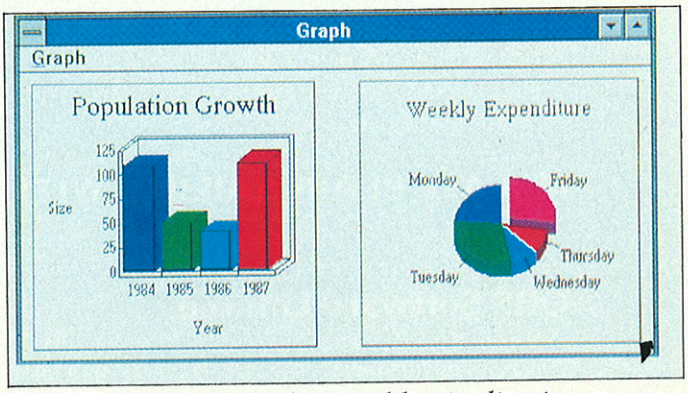


Figure 4 - A ChartBuilder Application

BLINKER™ TAKES CLIPPER™ OUT OF THIS WORLD

Blinker eliminates the need for complex overlay structures for both Clipper Summer '87 and 5.0

* Significantly reduces run-time memory requirements
* Accelerates your development cycle 3-10 times faster than any other current linker and even faster when linking incrementaly
* Dynamically overlays C and Assembler code sich as Extend.lib, dGE, FUNCky, other Clipper Add-ons and your own 'C' code
* Optimises your application with advanced profiling functions
* Includes memory defragmentation, 'burning-in' of Clipper environmental variables/serial numbers and creation of demonstration versions

IT'S ALL PLAIN SAILING WITH BLINKER

What the experts say

Here are just some comments from the 5000 happy users to date

"It's blinking fast!"

"Congratulations – we gained over 100k"

"It's so good, I went through all my programs and re-compiled/linked just for the pleasure of seeing all the free memory"

"It's fast! Handles overlays like a dream"

"Amazing – No more messing about with overlay structures!"

"Brilliant!... But my coffee breaks have been shortened!"

"Actually exceeded our most optimistic performance hopes! (Just for once)"

European sales and support:

QBS Software Limited

10 Barley Mow Passage, Chiswick, London W4 4PH

Telephone 081-994 4842. Fax 081-994 3441. BBS 081-747 1979

All known registrations for free update now shipped.

Blink Inc Limited, TCL Centre, Cwmbran NP44 3XR

Pens and Brushes control the appearance of shapes. Textpens provide a similar facility for text. There are seven Textpen attributes. The foreground and background colours of the text may be set, but the background text colour will only appear if the transparency attribute is set to `tm_Opaque`. Like Pens, the *Combination* attribute determines the logical drawing operation that will be used to write text into the drawing area. Finally there are attributes which control the font, text style and the height of the text. Each font is given a generic name (eg under Windows, `fs_FixSerif` is set to Courier), which prevents the use of platform-specific font names.

Graphical Objects

Here lies the heart of OGL, and the library provides a consistent way of manipulating graphical objects using member functions. The *TShape* branch in the hierarchy encompasses all rectangular shapes. These shapes are defined using the *Origin* and *Corner* coördinates to provide the corners of a bounding rectangle. Both the size and the position of such a shape may be changed by simply using the `SetOrigin` and `SetCorner` methods to change the origin and corner of the bounding rectangle respectively. OGL includes such diverse objects as lines, rectangles, and ellipses in its library of rectangular shapes.

Once an object has been created, it can be displayed or added to a list of graphical objects. All of the output ports mentioned above have a `Picture` field for this purpose. An object may be inserted into the list with the `Add` method. The complete picture will be displayed later when the `Picture^.Draw` method is invoked. The bounding rectangle may be used, in an interactive application, to determine whether the object has been selected. This is achieved using the `Contains` method which checks whether the mouse's coördinates lies within the bounding rectangle, and subsequently returns a valid pointer to the chosen object. The bar chart in Figure 3 is created as a *Picture*. Clicking the mouse on an axis adds a label to that axis. If one of the bars is clicked, a dialog box pops up, providing additional information on the selected bar. Figure 2 shows the code to respond to these mouse events.

As in ObjectWindows, OGL fully supports stream I/O using the `Put` and `Get` methods. Each object must be *Registered* to the application once. OGL provides a `RegisterOGL` function to register all of its object types. There is only one metafile format (OGL file format), and this handles the storing of pictures. It consists of a `TGraphSpace` header, followed by a `TPicture` object.

The design of OGL permits graphics applications to be written which are portable. OGL applications should run, unmodified, on any platform that supports OGL (presently only Microsoft Windows), although Whitewater has not given any indication to when other platforms will become available. The class structure works effectively, but I felt that using the `Associate/Dissociate` pair to write to ports was a little cumbersome. By harnessing the powerful features provided by the `TPicture` class, it is possible to create brilliant interactive drawing applications (just take one look at the superb ObjectDraw program) and it seems that OGL has been specifically designed for this purpose.

Controls	
AutoInc	Auto increment set & point counters when setting array data.
DataReset	Reset any/all of the data arrays.
DrawMode	Drawing mode (clear, draw, copy, print, write).
NumPoints	Number of points in data set.
NumSets	Number of data sets.
RandomData	Automatically generate random test data.
ThisPoint	Current point number (1-n).
ThisSet	Current set number (1-n).
Style	
Background	Graph background colour.
BottomTitle	Title at bottom of graph.
DrawStyle	Monochrome/colour drawing.
Foreground	Graph foreground colour (for text, axes etc).
GraphCaption	Graph caption.
GraphStyle	Graph style options (vary according to GraphType)
GraphTitle	Main title at top of graph.
GraphType	Graph type (Bar, Pie, Line, Area etc).
GridStyle	Horizontal and/or vertical grid.
Labels	Label on/off.
LeftTitle	Title on left of graph.
LegendStyle	Monochrome/colour legends.
LineStats	Line graph statistics options.
PatternedLines	Patterned lines on/off.
PrintStyle	Print style option.
SeeThru	'See-thru graph' option.
ThickLines	Thick line on/off.
Array	
ColorData	Colours for bars, pie-slices etc.
ExtraData	Additional style data.
GraphData	The data to be graphed.
LabelText	Label text strings.
LegendText	Legend text strings.
PatternData	Line/pattern style data.
SymbolData	Symbol style data.
XPosData	Independent X-variable data.
Technical	
CtlVersion	ChartBuilder version & copyright notice.
Help	Activate Windows Help.
ImageFile	File name for image output.
Picture	Picture handle for passing to Visual Basic Picture control.

Figure 5- ChartBuilder Properties

ChartBuilder Control

The other tool that I have been looking at is ChartBuilder Control which is an 'add-on' for Visual Basic (VB). You're probably already aware of VB's features, but here's a quick summary. VB is Microsoft's attack on the 4GL and application generator market for Windows. It provides a 'Forms' designer that enables windows, dialog boxes and menu bars to be interactively drawn at design time, before any actual code has been written. There are a number of 'controls', such as scroll bars, buttons and list boxes, which can be drawn onto the form by clicking the appropriate icon in the Forms designer toolbox that appears to the left of the drawing area. An applications programmer only needs to write code to handle the events generated by these controls. ChartBuilder Control (CB) is a custom control from Bits Per Second that can be added to VB's toolbox and enables graphs and charts to be drawn onto a form.

CB is supplied on both 3 1/2" and 5 1/4" formats and occupies about 680 KB of disk space. There is also a 95 page user manual. Installation is simply a matter of copying all the files on the distribution disk into an appropriate directory on your hard disk. A path to this directory should be added to AUTOEXEC.BAT. Once this has been done, a VB project can incorporate CB by including the GRAPH.VBX file in the project list using VB's **Add File** menu option. The ChartBuilder icon will then appear in VB's toolbox.

CB behaves like any other tool in the toolbox. For instance, each graph or chart on the form may be given a unique name using VB's *CtlName* property. The size and position of the graph can also be changed. In total, CB gives VB 35 new properties for manipulating the new control. Figure 5 lists all the properties that CB provides.

Interactive Design

Clicking on the ChartBuilder icon produces a 2D bar chart with five bars. When this control is re-sized, repositioned or one of its properties is changed, the graph is redrawn using random data. The first step needed to produce the Pie chart in Figure 4 is to change the *GraphType* property. On selecting '3D Pie', CB draws a 3D pie chart with five segments. The title of the graph can be changed using the *GraphTitle* property. In addition, labels may be assigned to each segment with the *LabelText* property. That's all the steps needed to produce a stunning 3D pie chart. Other graphs, such as the 3D bar graph, could be similarly produced.

Customising Graphs

CB provides several types of graphs which are determined using the *GraphType*

property, as shown above. There are 2D and 3D pie charts; 2D and 3D bar graphs; Line, Log/Lin and Polar graphs; Area graphs; Scatter graphs; Gantt and High-Low-Close graphs (I didn't know what these were either, but they certainly look impressive).

The actual appearance of a graph can also be changed using the *GraphStyle* attribute. On a Line graph, CB enables symbols to be used as markers for each data point. In addition, the data may be represented as individual sticks or they may be joined together with lines. In fact, all three of the above properties could be combined arbitrarily to produce the desired effect. Taking this one step further, CB even provides a number of built-in line patterns and marker symbols. In effect, it is possible to create a graph that you can call your very own.

Interfacing with code

It is all very well being able to design pretty graphs interactively, but what use is a graph if you can't display meaningful data on it? This is where some real coding is required. Since most graphs are plotted with *x* against *y*, CB uses a single array (*GraphData*) to hold the *y* values. The index into this array determines the corresponding value of *x*. Unfortunately, a major limitation of VB prevents property arrays from being directly accessed by a programmer. CB overcomes the problem by holding the current value of the index in the *ThisPoint* property. However *ThisPoint* cannot be used as a counter (eg in a `For` loop) so, normally, it would be manually set in order to access each element in the *GraphData* array. Luckily, the *AutoInc* property provides a short-cut by automatically incrementing the *ThisPoint* index each time the array is written to.

```
Graph.AutoInc = 1
For x% = 1 To 5
  Graph.GraphData = x
```

Since the lower bounds of the array is fixed (1 - n), the upper bounds is determined by the *NumOfPts* property, which controls how many data values are present. There are actually eight arrays in CB and any of them may be changed using the above procedure. Of course, it will only work if one array is being set. In Figure 6, the code to draw the Bar graph needs to set both the *GraphData* and the *LabelText* properties. In this case *AutoInc* cannot be used, since *ThisPoint* would be wrongly incremented twice during each iteration of the `For` loop. To overcome this, *AutoInc* is disabled, so that *ThisPoint* must be incremented manually. The *GraphData* array may be reset quickly using *DataReset*.

ChartBuilder can be mastered in fifteen minutes (trust me!)

When drawing 3D bar graphs it may be necessary to display more than one set of data values using the Z-axis to create the illusion of depth. To achieve this, CB provides the *NumSets* attribute which determines how many sets of data are being graphed. Each set may be identified using *ThisSet*. This produces a multi-dimensional array, where each *ThisSet* element contains its own copy of the eight property arrays.

The *DrawMode* property should be used whenever it is necessary to control when and how a graph/chart should be displayed. Altogether, there are seven drawing modes for clearing, drawing, copying to clipboard, printing and writing to file. The printing mode enables the entire graph to be printed on its own, excluding the form. Unlike VB's `PrintForm`, the graph is produced at the highest resolution that the printer can offer. Either a metafile (.WMF) or a bitmap (.BMP) can be written to disk, although it is not possible to read back a graph that has previously been written.

CB works by providing a link between Visual Basic and the Graphic Server which is a separate program that actually handles all the graph drawing. As Windows enables multiple applications to be run simultaneously, the Graphics Server process is capable of supporting multiple clients.

CB is a welcome addition to VB's toolbox. As the new control operates exactly like VB's own controls, graphs may be added to forms in exactly the same way that a Button would be added. Interfacing with the new control operates identically to VB's own code interface. In fact, CB fits so well into VB's environment, that you may wonder how you ever survived without it.

ObjectGraphics and ChartBuilder are two products that tackle the problem with the Windows GDI in two very different ways. On the one hand, ObjectGraphics offers the developer a versatile tool that provides a complete object-oriented interface to the GDI, while on the other, ChartBuilder concentrates on giving the developer a tool that enables applications using graphs and charts to be produced in the shortest possible time. The learning curves vary considerably - to write even a simple ObjectGraphics program requires a lot of background information and it took me almost two full days to understand the basic concepts. ChartBuilder can be completely mastered in 15 minutes (trust me)! So if you need a powerful tool for building interactive graphics applications then ObjectGraphics is the one for you. However, you may simply need a tool for presenting business graphics in your applications, in which case, ChartBuilder is the most appropriate choice.

EXE

ObjectGraphics is distributed in the UK by Neow which is currently offering it at a special price of £59 (usual price £149). The Pascal library source code is also available at £29. Neow can be contacted on 0628 668334. Whitewater may be contacted on 0101 708 3283800.

ChartBuilder Control costs £99 and is distributed by Bits Per Second Limited on 0273 727119.

```
Sub DrawGraph_Click ()
  Static BarCount As Integer
  Static i As Integer
  Static j As Integer

  Randomize                       ' Init random number generator
  BarCount = Int(4 * Rnd + 4)     ' Generate number in range 4 to 8.
  BarChart.NumPoints = BarCount   ' Set number of Bars.
  BarChart.AutoInc = False        ' Disable AutoInc facility .
  For i = 1 To BarChart.NumPoints
    BarChart.ThisPoint = i        ' Move to next position in graph array
                                  ' Construct Year label.
    BarChart.LabelText = Str$(StartYear + i)
    BarChart.GraphData = Int(100 * Rnd + 20)
  Next
  BarChart.DrawMode = Chart_Draw ' Draw Bar Chart

  PieChart.NumPoints = PieCount   ' Set number of Pie segments.
  PieChart.AutoInc = True         ' Automatically increment GraphData array.
  For i = 1 To PieChart.NumPoints
    PieChart.GraphData = Int(100 * Rnd + 1)
  Next
  PieChart.DrawMode = Chart_Draw ' Draw Pie Chart
End Sub
```

Figure 6 - Code to draw Pie Chart and Graph

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++	£199
Turbo C++ 2nd. Edition	£55
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£69
Pascal Professional	£129
Pascal for Windows	£99
ObjectVision	£79
Paradox 3.5	£335
Paradox Engine	£199
Paradox SQL Link	£215

AUTHORISED

DATABASES

Advanced Revelation	£795
Advanced Revelation -	£695
askSam	£255
askSam LAN	£579
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
DataEase	£485
DataEase - LAN 3-PAK	£485
Dataflex	£449
Dataflex Multiuser	£1065
Knowledgeman/2	£645
Magic PC	£325
Open Access III	£449
Open Access III Compiler	£385
R:BASE 3.1	£515
R:BASE Compiler	£190
R:BASE 3.0 LAN Pack	£645

DBASE

Clipper 5.0	£325
dBASE III PLUS	£395
dBASE IV	£495
dBASE IV Developer's Ed.	£695
dBFast/DOS	£255
dBFast/PLUS	£255
dBXL	£165
dBXL LAN	£389
Force	£449
FoxBASE+	£295
FoxBASE+/386	£425
FoxPro	£515
Vulcan	£320

OBJECT-ORIENTATED

Smalltalk/V	£65
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

CASE TOOLS

EasyCASE Plus	£190
EasyCASE Plus Prof.	£255
EasyCASE Prof. Upgrade	£80
Visible Analyst	£385
Visible Analyst Workbench	£2460
Visible Analyst Workbench	£6995
Visible Analyst Workbench	£1295
Visible Dictionary	£595
Visible Prototyper	£595
Visible Rules	£595

VERSION CONTROL

MKS RCS 5.1	£165
MKS RCS 5.1 - 5 User	£645
PVCS Professional	£319
PVCS - 5 User	£1345
SMS	£395
Sourcerer's Apprentice	£175
Sourcerer's Apprentice	£429

TEXT AND UNIX TOOLS

MKS AWK	£65
MKS AWK DOS & OS/2	£129
MKS Programmer's Platform	£429
MKS Toolkit	£165
PolyAWK	£129
PolyAWK Toolkit	£145
SpellCode	£65

TRANSLATORS

BASTOC	£515
BAS_C Commercial	£579
dBx/dBPort	£389
dBx/dBPort with source	£645
FOR_C	£359
Pascal to C Translator	£99

Microsoft

BASIC PDS 7.1	£219
Visual BASIC	£99
C 6.0 Compiler	£210
C 6.0 & Windows SDK	£349
COBOL 4.0	£345
FORTRAN 5.1	£199
MASM 6.0	£65
MS-DOS 5 Upgrade	£59
Pascal Compiler	£149
QuickBASIC	£50
QuickC	£50
QuickC/Assembler	£85
QuickPascal	£50
Source Profiler	£49
Windows 3.0	£65
Windows S.D.K.	£260

Authorised

EDITORS

BRIEF for DOS & OS/2	£199
BRIEF upgrade	£60
BRIEF with dBRIEF	£275
BTags	£35
Cheetah	£159
dBRIEF for dBASE/Clipper	£99
dBRIEF for Paradox	£99
dBRIEF for R:BASE	£99
EDT+	£190
EMACS	£210
Epsilon	£129
KEDIT	£99
MKS Vi	£99
Multi Edit Professional	£115
Norton Editor	£50
SPF/PC	£159
SYNDIE	£319
VEDIT PLUS	£120

LINKERS

.RTLink	£190
.RTLink/Plus	£319
ALINK	£129
Blinker	£175
Plink w/LTO	£359

MAKE UTILITIES

MKS Make	£99
MKS Make - 5 User	£320
PolyMake	£129
PolyMake - 5 User	£485

PARSERS AND LINTS

MKS Lex & Yacc	£165
PC-Lint	£90

PROFILERS

.RTLink/Plus	£319
CHARGE	£65
Codesifter	£79
Inside!	£85
PC Metric	£129

DEMO BUILDERS

Dan Bricklin's Demo II	£190
Instant Replay III	£99
Instant Replay Prof.	£385
Show Partner F/X	£259

MEMORY MANAGERS

386 MAX Version 5	£85
Above Disk	£79
Blue Max	£100
HEADROOM	£85
HEADROOM Network Extens	£39
HI386 Complete	£65
MOVE'EM	£59
QEMM-386	£65
QEMM-50/60	£65
QRAM	£55
Turbo EMS	£65

DOCUMENTATORS

4c for Brief	£79
ABC Flowcharter	£190
ASMFLOW	£129
C-Clearly	£85
C-Doc	£120
Clear for C	£129
Clear for dBASE	£129
dAnalyst for C and C++	£190
dAnalyst for Clipper	£190
dAnalyst for dBASE III	£190
dAnalyst for dBASE IV	£190
dAnalyst for FoxBASE +	£190

FREE
Microsoft Flight Simulator
WORTH £45 WITH EVERY PRE-PAID ORDER OVER £100*

EasyFlow	£99
Flow Charting III	£165
SourceDoc	£190
The Documentor	£190
Tree Diagrammer	£65

DISASSEMBLERS

Sourcer 486	£85
Sourcer w/BIOS pre-proc.	£110

E-MAIL

cc:Mail	£450
cc:Mail EXPAND	£385
cc:Mail Remote	£190
DaVinci eMAIL	£645
DaVinci eMAIL Combo	£965

CODE GENERATORS

DATABOSS	£395
Matrix Layout	£195
ObjectVision	£259
ObjectVision for Windows	£259
PRO-C	£259
R&R Code Generator	£99
UI2 Developer's Release	£385
UI2 Touch & Go	£259
Zachary	£259

DEBUGGERS

Periscope 286/386 Pod	£389
Periscope I OK	£319
Periscope I/512K	£385
Periscope I/MC	£449
Periscope I/MC 512K	£449
Periscope II	£145
Periscope II-X	£129
Periscope IIw/Switch	£145
Periscope IV (16 MHz)	£775
Periscope IV (25 MHz)	£900
Sherlock	£129
SVS DBG	£210

MULTI-TASKING

DESQview 2.2	£85
DESQview 386	£125
Dr. Switch-OnCall	£65

BATCH ENHANCERS

Batcom	£39
Beyond.BAT	£65
Command Plus	£85
ProKey Plus	£85
The Builder	£99

TERMINAL EMULATORS

BLAST II	£195
COTERM/100	£145
COTERM/220	£160
COTERM/220	£160
COTERM/4010	£165
EM 320	£199
EM 4010	£230
EM 4105	£330
EM 4105 Network	£345
ICE.10	£190
SmarTerm 125	£190
SmarTerm 2392	£65
SmarTerm 240	£225
SmarTerm 320	£129
SmarTerm 340	£225
SmarTerm 400	£115
SmarTerm 4014	£145
Smarterm 470	£190
ZSTEM 100	£65
ZSTEM 220	£99

PROFESSIONAL SHAREWARE FOR DEVELOPERS

ASM Library (20)	£40
C/C++ Library (68)	£66
dBASE/Clipper (87)	£100
DOS Utilities (37)	£40
Novell Netware (54)	£66
Turbo Pascal (38)	£40
Windows (64)	£66

(Number of diskettes)

*All prices exclude V.A.T. U.K. carriage included. ***One offer at time of purchase, per order at published prices.*** VISA or ACCESS accepted.
Credit to approved accounts.
SUMMER '91 CATALOGUE
Call for your free copy of our 160 page catalogue, TOOLBOX.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

CIRCLE NO. 216

Exceptional Programming

Exception handling methods are often scorned by language designers, who think they are a back door way of using GOTOs. But David Cooper thinks they have a lot going for them.

Exception handling tends to be neglected by high-level language designers. Exception handlers rely on passing the flow of control of a program to labels that are dynamically defined. Language designers scent danger, and as a result have unjustly neglected the techniques. The C language community especially has suffered from this. I believe exception techniques are a valuable tool; I'll try to explain why. I'll also take a detailed look at what C++ has to offer.

When might you use an exception? One use is for error handling. Suppose you are writing a program library. Each call to the library can result in success or failure. You need to pass error conditions that arise back to the caller.

How would you do this? One technique is for each procedure to return a value which indicates success or failure. The trouble with this is that the error might arise in a deeply nested procedure call. Tradition and custom point to a number of *ad hoc* solutions. Sometimes library designers use a global error number which must be checked after every call, or they make the library procedures return an error number. Once a non-zero value has been produced, some sort of corrective action must be taken before the library is used again. The inconvenience of this approach is that programmers must check this error number after every call to the library. Not only is this prone to omission, but so much error checking obscures the structure of the program. The writer of the library has the same problem. Control must return to the caller immediately the error is discovered without the library making more internal calls which might lead to further errors.

Exception techniques provide a neat way of dealing with this. Each error that the library can report is associated with an 'exception'. Before using the library, the user specifies an *exception handler*, which is a program label where control will be transferred to if an exception occurs. When the library discovers an error it *throws* an exception. This causes the flow of control to leave the library and jump to the user's exception handler, which usually contains code to recover from or report the error. The relation of an exception handler label to an exception is defined at run-time.

```
class math_err { /* ... */ };
class overflow_err: public math_err { /* ... */ };
class underflow_err: public math_err { /* ... */ };

void my_func(void){
     try {
          // try to excecute the statments inside this block.
     }
     catch (overflow_err e){
          // only come here if overflow reported.
     }
     catch (underflow_err e){
          // only come here if underflow reported.
     }
     catch (math_err m){
          // come here if other maths errors reported.
     }
}

void raise_power(double *x,int y){
     // do normal calculations
     // ...
     // only do this if overflow error detected
     throw overflow_err("**",x,y);
}
```

Figure 1 - A skeleton C++ exception handler

This is illustrated by the example in Figure 1. I've used the elegant C++ syntax. The example handles either of two exceptions: `overflow_err` or `underflow_err`. It shows how you define an exception handler and throw an exception using C++. The `try{...}` construct in Figure 1 is called a *try-block* and represents the normal flow of control. The `catch{...}` construct is called a *handler*. When flow of control enters a try-block, the corresponding handlers are associated with the specified type of exception for the duration of the try-block. An exception is thrown by a *throw-point* (see the function `raise_power()` in Figure 1).

The program in Figure 1 behaves as follows.

- If no exception is thrown, the whole of the try-block statement will be executed and none of the statements in the handlers will run.
- If an exception of type `overflow_err`, `underflow_err` or `math-err` is thrown during the execution of the try-block (or from any functions called within the try-block) the flow of control will branch to the corresponding handler and the execution of the try-block will be aborted.

Look Before you Leap

An essential part of an exception handling mechanism is the ability to perform a non-

local jump. A non-local jump is like a goto, but with a destination label which is variable and could be inside a different function, possibly in a different module. You cannot tell at compile time where control will go.

How are non-local jumps represented in high-level languages? Some language designers allow the programmer to declare a label inside a function as public, just like any other variable, thus making it possible to export the location of the label. The program can then branch to this label at any time. The problem with this is that it is difficult to define just what is meant by jumping to a label that is half way through a function. If the function has parameters, just what value would the parameters have? What values should local variables have? When it performs a return statement, where should it go?

It's much easier to give a sensible definition of non-local jumps if you impose a restriction in terms of dynamic parent-child relationships on the function call stack: a function may only perform a non-local jump into an ancestor function. Program flow passes from the currently executing function to the ancestor function containing the label. Any functions on the call stack between the these two functions are aborted.

This is still ambiguous. A function may be called recursively, in which case several invocations of it may appear on the call stack. Clearly a non-local label must specify more than just a program address: it requires information about which invocation on the call stack this label relates to.

Non-local jumps do not translate to straightforward jump instructions. When the non-local jump is executed, the stack needs to be 'unwound'. As a minimum, this requires that the stack pointer and stack frame pointer (or whatever methods the language uses to keep track of local variables) are tweaked so that they point to the bit of stack containing local variables belonging to the ancestor function, so its execution can continue from the exception handler label (see Figure 2). Thus non-local labels have to record at least sufficient information to achieve this. They may also have to record more: for instance, the value of register variables.

A problem with non-local jumps is that they leave the execution threads of intermediate functions hanging in the air. This can cause all sorts of problems if these functions have acquired system resources meaning to return them when they finished. Ideally some support should be provided for coping with this problem. C and Ada provide no such support, but C++ allows a very elegant solution which I describe below.

High-level language exception handling works by allowing the program to define a run-time connection between a non-local label variable (designating the entry point of the exception handler) and a specified type of exception. When an exception is thrown, a non-local jump is performed to the exception handler.

As any Whitehall mandarin would tell you, it's often best not to make mistakes public

The program can redefine the exception handler for a particular exception at any time. It is good practice to ensure that regions where the exception handler is redefined (for instance the duration of a function call) are nested at run time and to ensure that the previously defined handler is restored when flow of control leaves the region. C++ and Ada both enforce this practice.

If an exception is thrown for which no handler has been defined, the program executes a default handler which will normally terminate it.

Quashing Exceptions

So far, I've concentrated on exceptions as a way of escaping from the immediate context by making errors public, where a more informed decision can be taken. But as any Whitehall mandarin would tell you, it's often best not to make mistakes public: if you can patch up the mistake and continue a lot of thought is saved all round. Similar considerations can apply in a program.

Consider the problem of handling the Break key from a program running under DOS. This is a situation where the use of exception handling is natural. Unlike more sophisticated operating systems, DOS handles the Break key synchronously, which makes raising an exception straightforward: the Break key has no effect on the running until it next calls DOS.

If the programmer doesn't bother to define a Break key exception handler the program should be terminated when the Break key is pressed. However, a more robust program might wish to ignore the Break key exception altogether. This implies that the program would wish control to return to the throw-point without unwinding the stack and aborting any presently running computations. I'll call this *quashing* an exception. Implementations that allow this are said to conform to the *continuation* model of exception handling.

Should quashing be part of an exception mechanism? The ANSI C signal handling mechanism allows it, but C++ gives thumbs down (and so does Ada). These latter only support the *termination* model of exception handling.

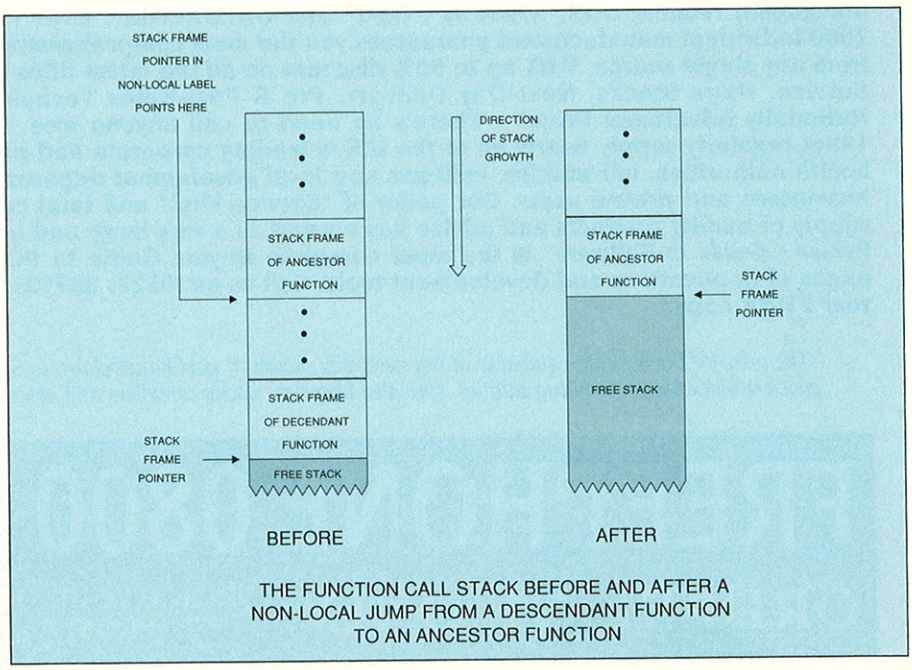


Figure 2 - The Stack before and after a non-local jump

CIRCLE NO 217

WE SUPPORT THE ETHICAL USE OF SOFTWARE. TO REPORT COPYRIGHT VIOLATIONS CALL F.A.S.T. ON (0628) 660377 NEW! REACH US ON COMPUSERVE AT MAILBOX 100015, 3570

E YOU WON'T
R RANGE OF
ANYWHERE †
ALLY ADVERTISED PRICE*

d excellent service which distinguishes order software suppliers." May '91

What Makes Us Different

- ✓ FREEPHONE 0800 Orderlines
- ✓ FREEPOST Mailing Address
- ✓ 24-Hour FAX Orderlines
- ✓ Up to 50% Discount On Over 15,000 Software Titles
- ✓ Pre-Sales Consultation
- ✓ Post-Sales Technical Support
- ✓ Unbeatable Prices
- ✓ Next Day Delivery
- ✓ Huge Inventory
- ✓ Knowledgeable Sales Staff
- ✓ VISA, ACCESS & Mastercard Accepted
- ✓ No Credit Card Charges Until Shipping
- ✓ Corporate PO's Welcome
- ✓ Educational & Volume Discounts
- ✓ Latest Versions Guaranteed With Full Documentation & Manufacturers Warranties
- ✓ Over 6 Years Experience Providing The Very Best In U.K. and U.S. Software

Please Specify Media Size When Ordering
Goods Are Not Offered On A Trial Basis
All Prices Exclude Carriage & VAT
Prices & Specifications Subject To Change Without Notice
† The products you see listed in this advertisement represent a small sample of our full range
All Trademarks & Registered Trademarks Are Hereby Acknowledged

Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE

* PRICE PROMISE

If you find an identical software product advertised in the national computer press by another mail order company based in the U.K. at a legitimate, current price - which is less than our advertised price - we will happily BEAT that price and SAVE YOU MONEY! We will also beat other suppliers with our friendly and efficient service and very wide range of brand-name, quality software. Since 1986 we've served 1000's of customers with this same promise. Discover today why we're one of the fastest growing companies in the industry. To all of our current customers - Thank you for your support, we really do appreciate it!

Access
MasterCard
VISA

Mail to: Software Paradise, FREEPOST, Caerphilly CF8 1ZZ

- FREEPOST TODAY -

NO STAMP NEEDED

Please rush me my FREE copy of "The Discerning Person's Guide to Software"

NAME

COMPANY

ADDRESS

POSTCODE

TEL & FAX

EXE 10/91

My view is that an exception is often a cry for help from a program module to its surrounding program environment; the module cries 'Something strange has happened. I don't know what to do about this situation; do you?'. The environment might very well be able fix it up, or might even view it as irrelevant. Or it might not. I want the exception handling mechanism to deal with both situations; the environment should be able to answer: 'I've fixed up the problem; go on as if nothing has happened'.

There are ways of managing this without explicit help from the exception mechanism. The service could first call a function to determine whether the problem can be fixed by the environment, and only raise the exception if the environment answers 'no' (see function `ask_then_throw` in Figure 3). The problem with this construct is that it is clumsier than the equivalent exception construct that uses exception quashing (illustrated by the function `just_throw` in Figure 3).

The lack of any in-built quashing mechanism in C++ means that special *ad hoc* measures have to be taken - for instance setting a flag meaning 'ignore the Break key' or some such trick. Getting rid of *ad hoc* flow of control constructs is one of the jobs of a high-level language. In this case Ada and C++ shirk their duty.

C++ Experiments

Exception handling in C++ is still at the experimental stage, and is not part of the formal language definition. It is described in the *Annotated C++ Reference Manual* (Ellis & Stroustrup, published by Addison Wesley). The major use envisaged by the language designers is for error situations: using C++ exception handlers to provide the user with hooks into a library package is inappropriate.

In C++, an exception is a data type, not a value. Resource clean-up during stack unwinding is handled very elegantly, through C++ constructors and destructors.

What are constructors and destructors? I'll give a brief explanation in case you aren't familiar with C++. C++ runs a programmer defined function when a variable is created, called a constructor. A constructor can do anything the programmer wishes; it's simply a function supplied by the programmer. Usually it would initialise the internal representation of the variable to a value depending on the arguments it gets called with (which are defined at the point the variable is declared).

Figure 4 gives an example of a class `complex` with a constructor (which is also called `complex`; constructors have the same name as the class). Not all classes have constructors, but if a class does, it gets run automatically when the variable is created. For variables that live in a stack frame,

An exception is a cry for help from a program to its surrounding environment

this is when their declaration is encountered in the program flow. The destructor gets run when the variable goes out of scope (there is no destructor in this example).

Destructors also get called when the stack is being unwound. So if you ever allocate a resource in a C++ program, do it inside the constructor of an automatic variable: if necessary declare a dummy variable. Do resource deallocation in its destructor. If you stick to that convention, when the stack is unwound resources allocated within aborted functions are de-allocated; the compiler handles resource deallocation for you. This is a very powerful mechanism indeed.

Information can be transmitted from the throw-point to the handler using a function call-like interface which is type-safe and general purpose. If you look at the example in Figure 1 you will notice that the throw-point is typed. So is the handler. So the exception raised by a throw-point will only be caught by a handler declared with a matching class.

An exception handler is allowed to re-raise the same exception by using the `throw` keyword without a parameter. This ability to transmit information in a type safe way is another advantage that C++ exception handling has over either C or Ada.

C++ leaves the problem of distinguishing between different exceptions to the programmer and unfortunately imposes no standard convention. Relying on uniqueness of classes to differentiate between exceptions is not dependable. There is no guarantee that the classes used by two different libraries will be different: after all, an integer is quite a reasonable way to represent an exception. If two libraries do use integers to represent exceptions, any handler of integer type will catch both sets of exceptions and we are back to the same administrative problems that C has. Even if you invent `my_new_class`, and demand that nobody use this class except for handling exceptions that you throw, fresh problems arise. The Ada approach of distinguishing exceptions by using symbol addresses resolved at link time is much safer.

A further problem with C++ exception handling is the means whereby type matching between the handler and the throw-point is achieved; a run-time type comparison is performed. When a C++ exception is thrown, the C++ exception handler mechanism searches through the list of installed handlers dynamically performing a type matching check to see whether the declaration of the exception handler it has found matches the declaration of the throw-point. This requires that type information be available at run-time. The type matching algorithm has to check for conversions from derived classes to base classes and between pointer types. If a match is found, it transfers control to the handler passing it the data value from the throw-point. The complexity of this is not very large. Doubtless a compiler writer would think nothing of it. I, however,

```
// ask if OK to continue after an error, throw exception if  not

ask_then_throw(){
     // come here on error...
     if (ok_to_go_on(context)){
          // continue
     }
     else
          throw error(context);
}

/* throw exception, expecting to return if error fixable
 (can't  use  this  technique  in  C++,  so  this  example  is
hypothetical) */

just_throw(){
     throw error(context);
     // get here if only if error fixed
}
```

Figure 3 - Quashing exceptions

Which applications generator offers all of the following features?

- No Runtime or Royalty Payments for any generated C Code
- Fully integrated Multi-Version Dictionary and Database
- WYSIWYG Screen, Report and Menu generation
- Data Managers

  PRO-TREE
  Btrieve
  Oracle
  Paradox
  C-TREE
  C-ISAM/INFORMIX
  dBase III

- Operating Systems

  DOS - Single or Multi-User
  UNIX
  XENIX
  QNX
  AIX
  Ultrix
  More environments on the way!

- Style of C Code

  K & R
  ANSI
  Proto

- Support/training

  East Anglian base
  On-site training available
  Modem support avaialable

PRO-C

The Open Approach.....

For further details, contact ITEL on 0206 577376 or 0902 763786 or write to:

ITEL
PO Box 1229
Colchester
Essex CO3 3LN

ITEL
4 Wychbury Road
Merry Hill
Wolverhampton WV3 8DN

would be reluctant to burden smaller embedded systems with this sort of run time overhead.

On balance, the proposed C++ exception handling mechanism has some superb ideas, and the notation is excellent. Unfortunately the C++ scheme leaves too much to the run time support and fails to guarantee discrimination between different exceptions.

Wish List

I have not discussed the use of exceptions in an asynchronous multi-tasking environment, since the facilities available tend to be very dependent on particular multi-tasking executives. The designers of C++ have been careful to define their exception handling mechanism so that it can be implemented reëntrantly allowing it to run in multi-tasking systems. However, the C signal and non-local jump support is not reëntrant.

This is my wish list for high-level language handling of synchronous exceptions:

- Low overhead of program space and time when exception occurs.
- Exception names administered at link time (at the earliest).
- Exception handlers have the option of quashing an exception.
- Arbitrarily complex algorithms can be run by the exception handler before it decides whether to unwind the stack.
- Resource clean up handled automatically.
- Type safe transmission of arbitrary amounts of information from the throw point to the exception handler.

I suspect that the use of exception handling techniques has declined with the widespread acceptance of high-level languages that hinder their use. Language designers are now beginning to make amends. The designers of C++ have created an excellent notation for taking advantage of these techniques; with minor changes they could make it very attractive to use.

EXE

```
class complex {
     double re,im;
public:
     // constructor for class 'complex', taking 2 real values
     complex(double r, double i){re=r; im=i;}
};

// declare a complex variable called 'i'
complex i = complex(0,1);
```

Figure 4 - A C++ Constructor

David Cooper works as software manager at Coherent Research Ltd, developing distributed embedded systems. He has over ten years experience of realtime software development.

TopSpeed V3.0 - C++ and Pascal

Following on from Richard Pickard's review of the new TopSpeed development system from Jensen & Partners, Paul G Smith takes a look at the Pascal and C++ components.

In this article I concentrate on the C++ and Pascal compilers and libraries; for information about the development system, the debugger, and Modula-2 language support please see Richard Pickard's article *(.EXE September '91)*.

The TopSpeed system, which includes Modula-2, C, C++ and Pascal compilers, plus an Assembler, is built on a set of common foundations. The development system and debugger are common to all languages; the code generator is shared; programs can be built that contain components from any or all of the languages; many of the libraries contain features that are equivalent in all the programming languages. The system is provided on a component-by-component basis; you can buy one language, or as many as you need. The TopSpeed environment is available for both MS-DOS and OS/2; I examined the MS-DOS version. Let's have a look at the Pascal support provided by TopSpeed.

Pascal

Pascal is dead. Roll on Son of Pascal++ (or is it daughter, brother, niece, or nephew of Pascal?). Don't get me wrong: I like Pascal and I have many years commercial experience using it. The trouble is, Pascal is showing its age seriously. Pascal has many dialects, each distinct. TopSpeed is both a superb implementation of a Pascal-based language and, at the same time, a pretty effective indictment of Pascal as a standard.

The Pascal language has reached the state of variegated maturity that BASIC achieved some years ago: it seems like the only assured common feature between all implementations is the name of the language. Nothing else is guaranteed to be consistent. This is because the original definition of the Pascal language is inadequate for large scale commercial program development, and because every vendor has chosen to implement compensating extensions in a different way. There was a time when I complained loudly about this diversity. Now, all I propose to do is report on it, highlighting differences, strengths, and weaknesses for others to make their own decisions. The time for traditionalism is over.

Fortunately, the authors of TopSpeed Pascal have made a real effort to conform to the International Standard (ISO 7185) for the Pascal language. ISO compatibility can be enforced by selecting a pragma option in the project.

Around this standard core, they add a large number of extensions ranging from type-secure separate compilation through aliases to object-oriented programming. Just to give you an idea of how many extensions they have implemented: the summary in the Pascal manual lists 22 (yes, twenty-two) categories of extension. I'll look into the extensions in more detail in a moment.

JPI also supports a 'Turbo compatibility' mode for the compiler, in which it relaxes some of its rules in order to help programmers convert programs from Turbo Pascal. This is not sufficient to compile any but the simplest (and most portable) Turbo Pascal programs, however, as the two Pascal dialects remain quite distinct. JPI provides a conversion utility to help programmers convert their Turbo source code to TopSpeed Pascal; it also provides some 'units' that contain functions compatible with many of those in the Turbo libraries. Should you wish to convert your Turbo source code to TopSpeed Pascal, the process is not likely to be effortless, but JPI has gone to a lot of effort to provide tools and documentation that will help you. The conversion process is strictly one-way.

TopSpeed Pascal: language

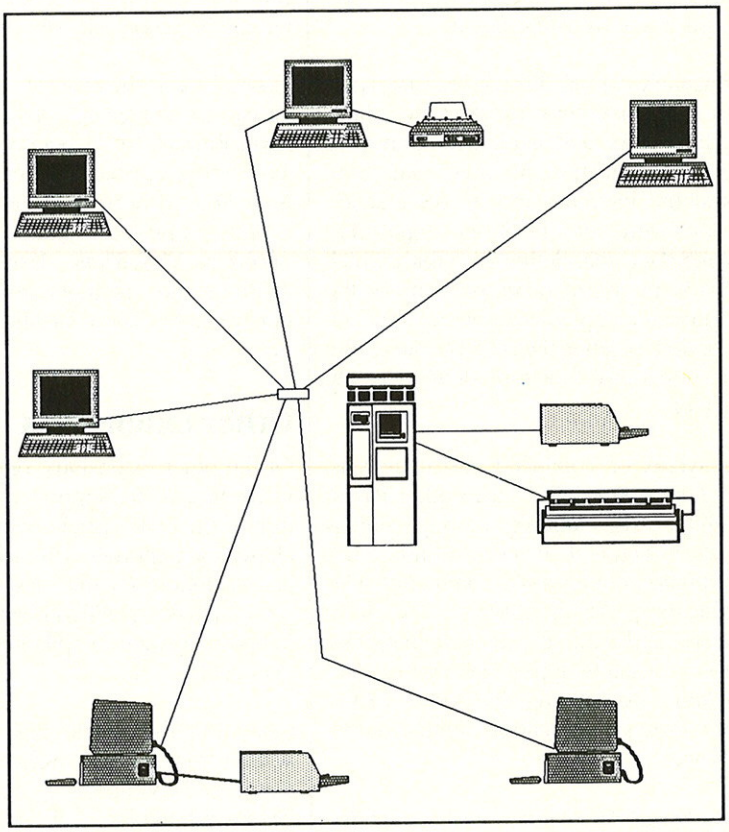
Unlike many other PC implementations this one does start from the base point of level 1 ISO 7185 compatibility. You can operate the compiler in a mode where the ISO standard is enforced. This is not the normal mode of operation, though: like most other dialects of Pascal, TopSpeed's implementation extends the standard language to provide those features that are essential for serious development work: things like support for separate compilation. TopSpeed also add a few special bonus features all their own.

The TopSpeed Pascal language reference manual lists twenty-two sorts of extension, including (this is a selection of the more significant ones): separate compilation,

```
INTERFACE UNIT comms;

CONST
  min_speed = 300;
  max_speed = 19200;
TYPE
  s_baudrates = min_speed..max_speed;
  s_parity    = (none, odd, even, mark, space);
  s_ports     = (Com1, Com2, Com3, Com4);
  s_channel_config = RECORD
    speed:  s_baudrates;
    parity: s_parity;
    port:   s_ports;
    END;

VAR
  channel:  s_channel_config;

PROCEDURE Init_Port(speed: s_baudrates,
                    parity: s_parity,
                    port: s_ports);
PROCEDURE Send_Data(sendData: string[11]);
          { "conformant string" param }
PROCEDURE Close_Port;

END.
```

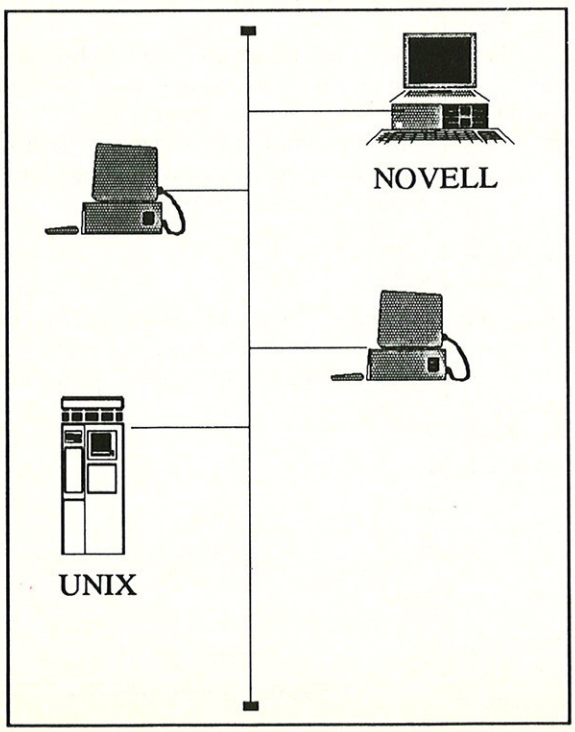
Figure 1 - A TopSpeed Pascal interface unit

SMOOTH SKATING BETWEEN DOS AND UNIX/XENIX



*** ICE.TEN ***

- Exact WYSE 60, ANSI, VT220 & VT100 terminal emulations, memory resident in 42K of DOS memory, giving instant switching between DOS and UNIX screens.
- Uses standard RS232 ports/cables
- Support for ANSI colour codes in all emulations.
- Fully programmable function keys
- Transparent and local print to PC's parallel port.
- Easy file transfer between PC and Host with an MSDOS style copy command.
- DOS applications can share UNIX printers.
- From £350 for an UNLIMITED no. of PC's linked to a single host.



*** ICE.TCP ***

- A DOS version of TCP/IP with the same exact terminal emulations as ICE.TEN
- Fast and simple file transfer.
- Can coexist with Novell and other DOS networks
- Drivers for most popular Ethernet cards.
- £195 for 1 PC, £995 for 10 PC's, £2995 for 25 PC's, £4995 for 100 PC's.

Custom Business Systems Ltd

TEL: 071 323 2297

MSDOS to UNIX experts.

CIRCLE NO. 222

alias definitions, based pointers, absolute variables, optional and structured type functions, procedure and function types, and object-oriented programming.

Separate Compilation

TopSpeed's model for separate compilation is based on that employed by TopSpeed's Modula-2 language. Although similar on the surface, it's quite different in detail to the 'UNIT' structure employed by descendants of the original UCSD Pascal compiler, and by Turbo Pascal.

A separately-compiled TopSpeed Pascal program is split into units. There are two kinds: interface units, and implementation units (note the distinction: in earlier dialects of Pascal the interface and implementation sections were portions of one larger unit). The interface units define everything which is made public to other portions of the program; the implementation units (which are optional) define the actual bodies of the procedures and functions defined in the interface.

The syntax of TopSpeed Pascal units is, therefore, a little different to other Pascal compilers. An example interface unit is shown in Figure 1. Its corresponding implementation unit would commence with the heading 'IMPLEMENTATION UNIT comms' and continue, after defining any private constants, types, procedures, and functions, by defining the bodies of the procedures and functions defined in the interface.

A TopSpeed Pascal unit is treated a bit like a big object-oriented class. When a unit is used by (or 'imported' into, in TopSpeed-speak) another Pascal compilation unit, it is declared via an IMPORT clause. As in Ada, it is possible to force the user to name the source unit of a given identifier like this:

```
<unit_name>.<identifier>
```

Although the TopSpeed model for separate compilation is different from what I'm used to, I like it. In large applications it serves to make the ownership of identifiers a lot clearer. The downside is that it makes Pascal source code even more long-winded than before (which won't impress the C fraternity).

O-O extensions

The object-oriented Programming extensions supported by TopSpeed Pascal go a bit further than predecessors like Turbo Pascal and Quick Pascal on the PC, and MPW and Think Pascal on the Mac. This means that there are syntactic differences between the OOP extensions in this and other Pascals. You can't just port your existing source code across.

TopSpeed Pascal's OOP extensions include class definition and implementation, single and multiple inheritance, aliasing of inherited identifiers, overriding inherited methods, support for both virtual and static methods (unlike Turbo Pascal, virtual methods are the default and static methods must be explicitly identified: full marks to JPI for this), and an IS operator for comparing the actual class of object instances.

Take a look at the example of a TopSpeed Pascal class in Figure 2. Object instances are normally stack-based, but pointer-based objects can be created and deleted using the New and Dispose built-ins. There is no support for protected and private members of objects. Virtual base classes, overloaded methods and operators, and pointers are not available. (These facilities are available in C++.)

Other extensions

Of the other extensions, one of the most interesting is the support of 'alias definitions', through which one can declare aliases for identifiers. This is most useful in the resolution of name clashes; it can also serve to make qualified identifiers (see my notes on separate compilation above) more intelligible.

One thing, that might confuse those converting from pre-ISO dialects of Pascal, is that nested comments are not permitted in ISO Pascal. In older dialects one could use the two forms of comment { ... } and (* ... *) independently, nesting one kind in the other. This allowed one to use the { comments as usual, and to then block out entire portions of code and comments using the (* comments, which is no longer possible.

Compiler directives (to be more accurate, 'pragmas') in the TopSpeed environment use a slightly different syntax to that familiar to those brought up on descendants of the original UCSD Pascal compiler. In the UCSD model, if the first non-delimiter character is a $ the comment is a compiler directive; in TopSpeed, if the first non-delimiter character is a # then the comment is a pragma, and if it is a % then the comment is a conditional compilation directive.

The compiler

There are two versions of the TopSpeed Pascal compiler: both are supplied with the TopSpeed system. One has a much larger symbol table than the other, and (at the expense of longer compilation times) can compile very large programs. The other compiler is the default. Both can be used from the DOS command line, as with the other components of the JPI TopSpeed development system.

Pascal Libraries

The TopSpeed Pascal system is supplied with a variety of libraries. The standard library set includes: *PasLib, PasDos, PasProc, PasRed* and *PasWin.* PasLib implements a number of standard facilities required by TopSpeed Pascal, including routines required by the compiler and many callable procedures and functions. The routines in PasLib are always available to your programs, because it is imported automatically by the compiler. Facilities provided include screen and file I/O, memory management and manipulation, floating point numerics, type conversions and string handling.

PasDos contains procedures that interface to the operating system. PasRed interfaces to the TopSpeed file redirection system, which is common to other TopSpeed languages. PasWin implements a text windowing system. PasProc implements a process scheduler with which you can build multi-threaded programs containing effectively concurrent process threads that communicate by means of semaphores. PasProc is also based on a standard TopSpeed foundation.

Source code for the Pascal libraries is available as an add-on JPI module, so you can understand how they work and, if you really have to, change them.

Source code for some extra O-O libraries is also provided, as a starting point for your O-O development. The two libraries are *Classes*, which contains a number of collection classes, and *Streams* which defines and implements stream I/O classes.

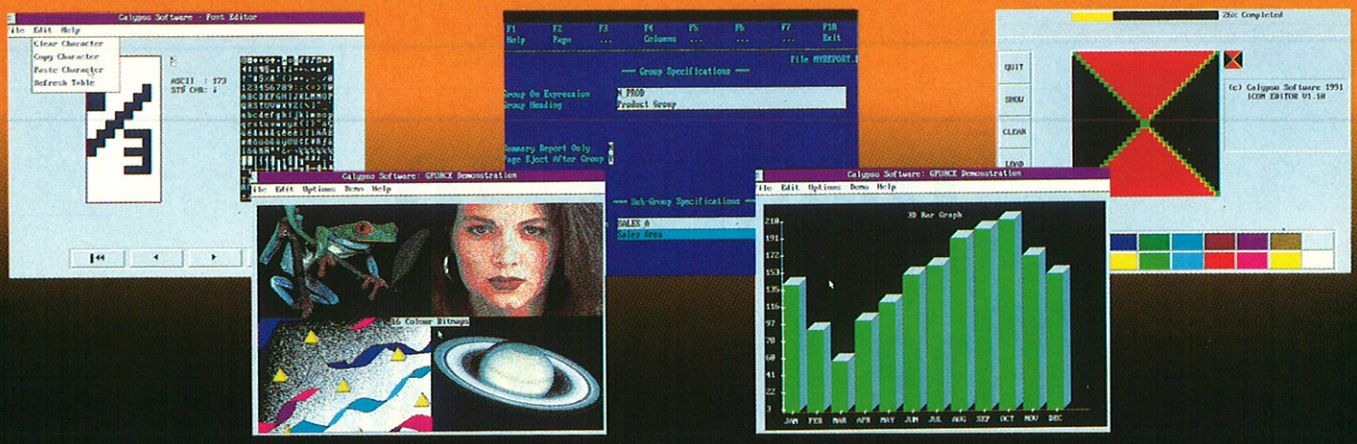
C++

TopSpeed C++ is, according to JPI, the first MS-DOS C++ implementation that fully conforms to this AT&T C++ language de-

```
CLASS connection;
  speed:  s_baudrates;
  parity: s_parity;
  port:   s_ports;
  PROCEDURE Open_Conn(thespeed: s_baudrates,
                      theparity: s_parity,
                       theport: s_ports);
  PROCEDURE Send_Data(sendData: string[11]);
  PROCEDURE Close_Connt;
END;
```

Figure 2 - A TopSpeed Pascal class

Spot the difference...



All of these screens have been taken from Clipper applications. The difference is that four of the screens use the GFORCE graphical user interface.
GFORCE provides a very fast Windows® like environment as a linkable library for the Clipper compiler.
GFORCE includes low level functions for pixels, lines, boxes, circles, fills, bitblits and high level functions for 3D buttons, pull down menus, dialogs, scroll bars, icons, bitmaps and fonts.
GFORCE retails for £185.00 including full documentation, Norton Guides, Quick library with complete Clipper source, graphical font and icon editors with Clipper source and ninety days technical support.
GFORCE works with Clipper® 5.01, Summer '87 and overlay linkers such as Blinker® etc.

Please call for a free demo disk.

QBS Software Ltd
10 Barley Mow Passage
London W4 4PH, ENGLAND
Tel: 081 994 4842
Fax: 081 994 3441

GFORCE™ Clipper Graphical User Interface

CIRCLE NO. 224

All you need to rewrite the rules of accounts software.

There are no 'can'ts' with dBFLEX software.

Available off-the-shelf in source code form, it offers no restrictions on how you may edit or extend the system to suit your company (or client).

dBFLEX integrates every financial module a business could need. You get the performance of other 'number crunching' packages, multi-user operation and the flexibility of standard dBASE databases.

This means you can easily link to other language versions including dBASE, Clipper and FoxPro products.

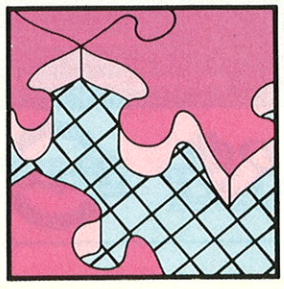
Your existing programs (or host of proprietary ones) can then be linked directly to dBFLEX - saving the time and the expense of developing customised solutions.

Best of all, these enhancements can be supported by the original authors of dBFLEX - Dataflow (UK) Limited - via a Technical Support agreement.

For the facts on dBFLEX, call **0784 454171** for the address of your nearest dealer. And join over 1200 companies who have already re-written the rules.

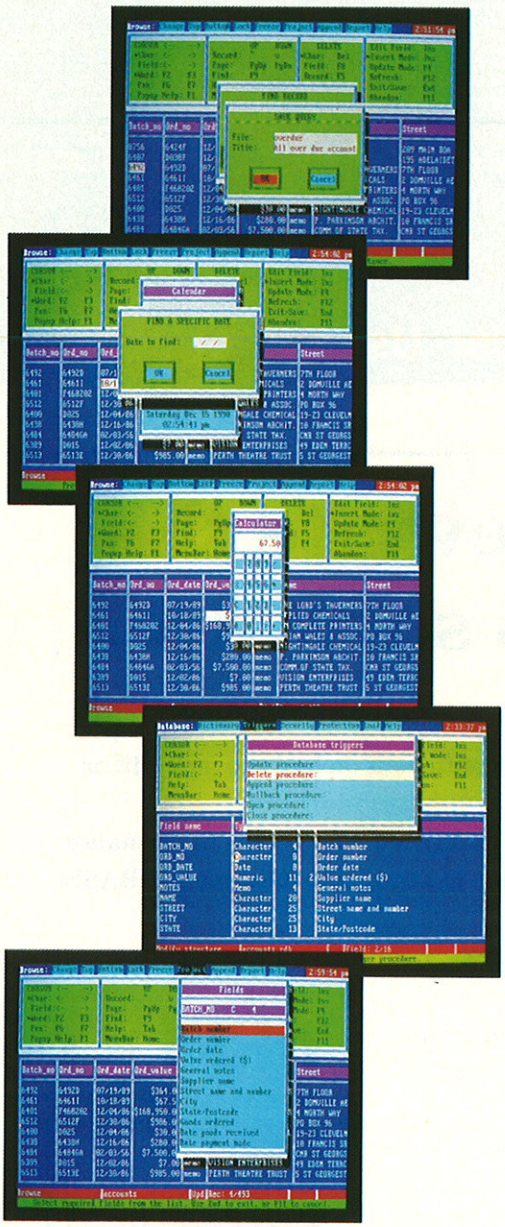
Dataflow (UK) Ltd, Dataflow House, Mill Mead, Staines, Middx TW18 4UQ

dBFLEX
BUSINESS SOFTWARE



CIRCLE NO. 223

Recital 7.0 Is The Only Choice.



- Advanced RDBMS/4GL
- Active Data Dictionary
- Object-oriented with pre-defined "application objects"
- Open client/server architecture
- Integrated pop-ups for calendar, calculator, choice lists
- Built in "Dynamic Compiler" New v7.0
- Pop-up time management scheduler New v7.0
- Relational multi-record table fields in forms New v7.0
- Split screen relational browse New v7.0
- Database, form and report triggers New v7.0
- Automatic record locking
- Supports 14 datatypes including memo, numeric, byte, integer, short, character, date, logical, float, real, packed, zoned, vaxdate, quadword
- Available now on 100+ VAX/VMS and UNIX computers (including full VAX cluster support)
- SQL (interactive and embedded)
- dBASE, FoxBase and Clipper compatibility (yes, even on VMS!)
- Terminal (including full color support) and keyboard independence
- Transparent connectivity to existing RMS and RDB databases on VAX/VMS
- Completely integrated development environment
- Powerful end-user tools

Recital
CORPORATION

dbSVP SOLUTIONS · VENDOR · PROGRAM

88open COMPATIBILITY CERTIFIED

Recital Corporation LTD, South Bank Technopark, 90 London Road, London SE1 6LN, United Kingdom, Tel 071-401-2727, Fax 071-633-9617
Recital Corporation Inc., 85 Constitution Lane, Danvers, MA 01923, Tel (508) 750-1066, Fax (508) 750-8097

CIRCLE NO. 225

finition (known as C++ version 2.1). I am not in possession of a standards verification suite, so I cannot prove or disprove JPI's claims, but certainly I had no problems running my tests through this compiler. Presumably, should the C++ standard that is eventually agreed differ from the present definition, JPI will amend its product to match the standard.

Because TopSpeed C++ is much more 'standard' than the Pascal compiler, there's less for me to say about it (I don't propose to repeat the language definition). However, there are a small number of extensions that are worth reporting on.

Optionally, nested comments are permitted (subject to a special pragma). Based (relative) pointers are permitted, which take up the same space as a near pointer except that the segment address is specified in the pointer declaration. Additional keywords (such as `cdecl`, `pascal`, `huge`, and `interrupt`) are available, for use as modifiers to declarations; the syntax of declarators is extended accordingly. Functions can be initialised, with the effect of defining in-line machine code. Lvalues can be cast. Pointers can be compared with like-sized integers.

The DOS environment variables are accessible, if they are declared as an optional third parameter to the main routine. These extensions can be disabled, with pragmas, if required.

If you have no C compiler installed, TopSpeed C++ is able to compile your C source files for you.

Interestingly, the C++ release notes report a small number of 'known problems', whereas the Pascal release notes report none (no C++ problems are particularly worrying, and you should be able work around them satisfactorily). Perhaps this is not surprising, considering the relative complexity of the C++ language.

Standard class libraries

Supplied with the TopSpeed C++ compiler is a full set of ANSI-standard C language libraries. In addition, JPI provides some C++ specific extras.

The Complex library implements complex numbers as a user-defined type, and implements the AT&T C++ release 2.0 complex number class. The biggest of the standard C++ libraries is the Streams library, which implements the full set of AT&T C++ release 2.0 stream classes. Also provided is a BCD number and arithmetic class.

JPI also includes, with the C++ compiler, a library of task and process management classes. These, although largely equivalent to the AT&T tasks library, are implemented on top of the process management architecture supplied with the other TopSpeed languages. Lastly among the standard libraries is the text windowing library, which is again equivalent to those provided with the other TopSpeed languages. As with the other languages, source code to the standard libraries is available on purchase of a separate module. Another add-on module contains the Rogue Wave class library - see separate box for a description of this.

Common Foundations

Being built on common foundations means that the various TopSpeed languages share many facilities, having such benefits as a common set of pragmas to control compiler and linker behaviour. However, naming conventions and data formats (such as those for strings) are not automatically the same. You have to use pragmas to enforce compatibility if you are developing programs that have components written in more than one language, or if you are writing a library in one language that is intended to be used from others.

The TopSpeed linker, in version 3.01, is notable for being able to strip out virtual functions that are never callable within a given program. This will considerably reduce the size of programs that inherit large classes but only use portions of them, and puts a feather in JPI's cap at the expense of some of its competitors.

The TopSpeed Tech Kit (yet another optional component that you can purchase) offers you a number of extra facilities, including: support for developers of MS-Windows software; segment-based overlay management for large programs and their data; dynamic link library support for MS-DOS as well as OS/2; additional documentation to help you link modules written in different languages; an Assembler; additional development tools such as a post-mortem dump analyser, a profiler, an object module disassembler, and a TSR that lets you monitor DOS function calls and their parameters as they happen; documentation for developing embedded systems using the TopSpeed system; additional technical documentation for more specialised aspects of the TopSpeed system; some extra goodies for the Modula-2 programmer, including TSR and RS-232 support.

Rogue Wave class library

The Rogue Wave class library, licensed by JPI from Rogue Wave Associates, is a collection of C++ classes that can be used as a foundation for your own projects. The Rogue Wave library does not attempt to provide an applications framework, as do libraries like Turbo Vision and MacApp. What the Rogue Wave classes do is provide a large set of building blocks, that can be used independently or combined as necessary.

The Rogue Wave library offers an impressive set of tools. These range from Smalltalk-like collection classes, through standard and B-tree indexed file operations, and string operations, to a very powerful selection of mathematical classes and functions.

The collection classes include many modelled on Smalltalk-80, and include classes like `Set`, `Bag`, `OrderedCollection`, `SortedCollection`, `Dictionary`, `Queue`, `Stack`, and so on. All can be made use of, in any new class, by inheriting the class `Collectable`. As well as the Smalltalk-like collection classes, generic collection classes like singly and doubly-linked lists, stacks, queues, and vectors are implemented. These are built using GENERIC.HPP, so they can use the currently defined mechanism for C++ parameterised types.

The File Space Manager (class `FileManager`) deals with the allocation and deallocation of free space within files; the `RWFile` class is provided to encapsulate standard operations on files. A `BTreeOnDisk` class, built on top of the `FileManager` class, implements disk-based indexed files using B-Trees as the indexing mechanism.

The string and character manipulation classes provide operators and functions for indexing, concatenating and comparing strings, and for extracting and assigning to substrings. The time and date classes define classes and functions for time and date calculations, comparisons and formatted I/O. The Mathematics portions of the Rogue Wave library include: vector and matrix classes, overloaded arithmetic and trigonometric operators for vector and matrices, mathematical functions for statistics, simulation, and modelling, classes for performing Fast Fourier Transforms, classes for solving linear simultaneous equations, random number generators, and least squares fit classes and functions.

Docs and Support

In general, I like the TopSpeed system. I had a few niggles, but these were mostly down to the documentation which, although extensive and detailed, isn't well enough indexed or cross-referenced for my liking. Although it's easy enough to find a given topic, looking up some detail by index is not so easy: the indexes are poorly formatted, and often I found myself looking through index after index. JPI should provide a cross-referenced master index to all its manuals.

I have some reservations about JPI's technical support. Maybe I caught them at a bad time. My query resulted from the fact that, without any warning or explanation (in the manuals or README files) the TopSpeed environment stamps on an interrupt (0x60) that my network card uses. This meant that outside the TopSpeed environment my application worked fine, whereas within it my application would fail to find the network interrupt. I telephoned JPI's help line. The gentleman who answered my call did not have the answer and promised to call back. He didn't. On my third call a different gentleman confirmed, in a seriously disinterested manner, that the TopSpeed environment does indeed stamp on interrupt 0x60, and also that it uses another interrupt. (He couldn't tell me what the other one was, and nor could he tell me why the interrupts are used or how to work around the problem.) I fixed my problem myself, by changing the network software's interrupt to 0x64. As I said, maybe I caught them at a bad time...

Wrap-up

JPI is very keen to convert Turbo Pascal users over to the TopSpeed environment; likewise, it is keen to convert users of Borland C++ and Zortech (now Symantec) C++. In both cases, I would suggest that although the TopSpeed environment has many strengths over its competition (and indeed is better in many respects) that's not on its own enough reason to convert unless you need the multi-language programming, the multi-threaded execution facilities, or other extra features of the JPI product. If you are up against the limitations of your existing C++ or Pascal development then yes, you should look at TopSpeed carefully. If you have not had any problems with your current development system, you should try to identify the benefits of converting before you do so.

The above qualifications apart: JPI's TopSpeed environment is very good indeed, and the Pascal and C++ implementations are both excellent. Every PC programmer should take a close look at this product. I was very impressed by it.

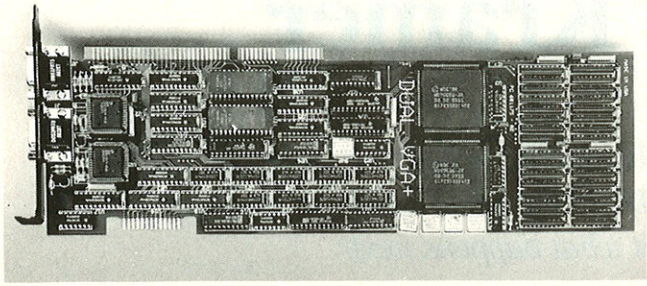
EXE

Paul G Smith has recently founded a new software house, trading as 'CommsTalk HQ', to develop and market a range of communications software development tools. He is not having a holiday this year. He can be reached via AppleLink and CIX as `pgsmith` *and on CompuServe (user id "100010,341").*

Paul was testing the following JPI TopSpeed modules: TopSpeed C++ compiler, TopSpeed C++ Library Source Kit, Rogue Wave C++ Class Library, TopSpeed Pascal Compiler, TopSpeed Pascal Library Source Kit, TopSpeed Environment and the TopSpeed Professional TechKit. Each module costs £59 (no discount for multi-module purchases, which seems a bit mean), available from JPI direct on 0234 267500.

MULTIPLE INDEPENDENT MONITORS
CAN NOW BE DRIVEN SIMULTANEOUSLY
FROM A SINGLE PC OR PS/2

THE DUAL VGA PLUS



★ UP TO 800×600 RESOLUTION
★ TWO INDEPENDENT SIMULTANEOUS DISPLAYS WITH OVERLAY CAPABILITY
★ 100% COMPATIBLE WITH IBM VGA/EGA/CGA/MDA VIDEO STANDARDS
★ 4 BOARDS IN ONE SYSTEM GIVE 8 SIMULTANEOUS DISPLAYS

NEW DRIVER! ALLOWS WINDOWS 3.0 TO BE EXPANDED ACROSS FOUR MONITORS

CEBRA COMMUNICATIONS LIMITED
26 LORNE PARK ROAD, BOURNMOUTH
TEL 0202 299048 FAX 0202 299192

CIRCLE NO. 227

EIFFEL: The Key to Software Quality

The Eiffel System: A unique combination of advanced object-oriented techniques, for the quality-minded Software Engineer.

The Language: multiple inheritance, genericity, assertions, disciplined exception handling, polymorphism & dynamic binding, strong static typing, deferred classes...

The Environment: automatic compiler, source level debugger, automatic documentation tools, graphical design tools, garbage collection...

The Libraries: data structures, parsing & lexical analysis, persistency, X-Windows, character windows, interfacing tools...

Hundreds of companies worldwide have grasped the opportunity to produce quality reusable software, and are using Eiffel for their development projects - Philips, Telecom Australia, Motorola, Hawker Siddeley, GEC-Marconi Research... Join them.

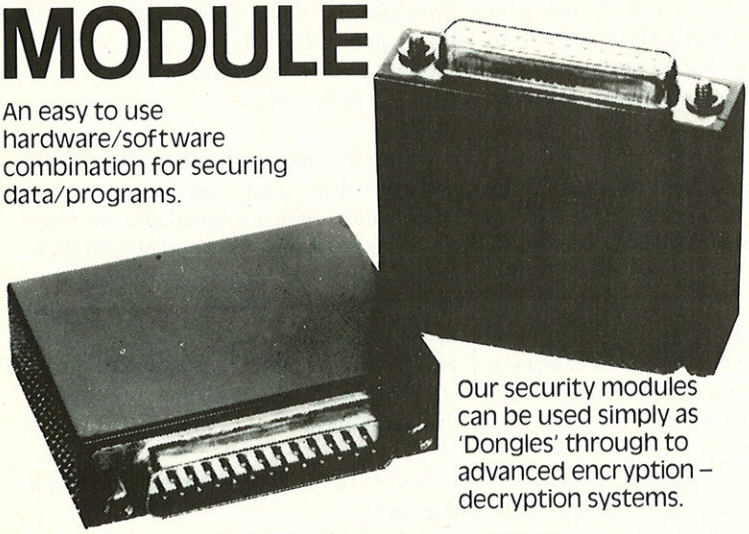
For further information contact:
Applied Logic, 9 Princeton Court,
55 Felsham Road, London SW15 1AZ;
Tel 081 780 1088, Fax 081 780 1941

Applied Logic

CIRCLE NO. 229

SOFTWARE SECURITY MODULE

An easy to use hardware/software combination for securing data/programs.

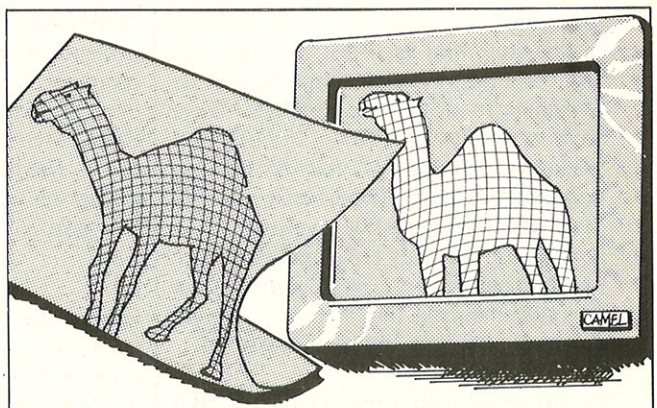


Our security modules can be used simply as 'Dongles' through to advanced encryption – decryption systems.

★ For IBM PC family (and compatibles)
★ Variants for other systems
★ Advanced construction – rugged and compact
★ Proven throughout the world
★ Transparent operation – user friendly
★ Driver software supplied for variety of applications

CTL
Control Telemetry of London
11 Canfield Place, London NW6 3BT
Tel: 071-328 1155 Fax: 071-328 9149

CIRCLE NO. 228



PLOT THE DIFFERENCE

PLOTVIEW

Is the simple, low cost alternative to paper plots. Able to preview plot images on a graphics screen within seconds,

PLOTVIEW allows paperless plotting and replotting – avoiding delays, improving efficiency.

PLOTVIEW

Has up to 25 separate plot files previewable simultaneously on screen and a zoom function which allows any part of the plot to be inspected in detail.

Easy to integrate, PLOTVIEW is ideal for system builders and can be customised to suit their individual needs.

Phone now for more details about PLOTVIEW: the versatile tool that can really make a difference to your productivity.

CAMEL SERVICES LTD.
Telephone OXFORD (0865) 512678

CIRCLE NO. 230

Kramer vs Kramer

It can't have escaped your attention that the PC is now ten years old. Modern machines are barely recognisable as the same machine released by IBM all that time ago, but what happens next?

Before the PC appeared, the world was a very different place. There were two kinds of computers - huge things which were worth more than some countries and which needed constant care from White Coats; and toy computers (which were toys because they were physically smaller than their manuals). There were no standards. A company would put together anything it felt like, without reference to what anybody else was doing, then when something new became possible the company would dump its products and start selling next year's model. Nothing ever worked properly. Software for all these machines was in short supply, and of universally low quality - why would anyone invest 100 man-years of work into a program which was going to sell 50 copies, and had a lifetime of six months?

At this time, IBM was seen as being staid and conservative, using committees of accountants to design its machines. There were two attitudes to IBM - 'Nobody ever got fired for buying IBM', and 'IBM over my dead body!'. Meanwhile, Microsoft was bubbling up by writing BASIC interpreters in a few KB for the likes of Commodore. The two companies were worlds apart; their announcement that they were to coöperate on a new computer stunned everybody.

I know people complain about the PC. But the fact is that the PC, almost single-handedly, put an end to this Babel. For the first time, here was a machine which was stable, fairly reliable, and was sure to be supported for years because all the specifications were public. The concept of the platform (with all the connotations of stability that the word implies) was born with the PC. Companies - indeed whole industries, were launched on the back of it. It took a collaboration of the kind that IBM and MS forged to make such a machine - it was full of compromises from beginning to end. The two companies together could find the middle ground between their established markets, and fill the gap perfectly. It is precisely because of the compromises that the machine works!

I'm not suggesting that the PC is a technically good machine. Commercial success, though, has never had anything to do with technical achievement - the PC was a standard, and even poor standards are better than none.

Now, ten years on, IBM and Microsoft have fallen out. After trying to bury MS-DOS and replace it with its own, proprietary operating system (and failing) IBM has flounced off to Apple ('They're not my best friends any more, you are!'), and Microsoft is doing everything it can to rubbish what IBM did, and explain why it has the best ideas.

I think IBM has shot itself in the foot. After watching far-eastern companies making their boxes far cheaper than IBM itself could, and perceiving this as lost dollars (instead of the investment in the whole industry that it clearly was) IBM is trying to build a migration path which locks people into its products. It won't work - part of the appeal of the PC was that it released customers from exactly this tyranny. Nobody with any sense is going to go back to single-vendor machines again.

On the other hand, Microsoft is doing an even worse job. In its Gadarene rush to show how much more clever than IBM it is, they released the half-finished DOS 4, Windows 3 (which was practically incompatible with W2); now it has released DOS 5 (which massages the disk in a whole new way - throw out your disk managers!), and Windows 3.1 is due soon (Ha!).

Windows 3.1 in particular contains a number of interesting innovations; True Type for example. Well, TT is a complete waste of time - everyone who needs it already has ATM, and MS has missed the boat. They've got New Technology - it was going to be the great white hope, but instead, it transpires that it's a hacked version of OS/2. And they have OLE - the thing that no fashionable program should be without!

It seems to me that if these things are so important, they should have been part of the original specification. None of it is technically very clever - most is just conventions rather than actual code. With the hype that surrounds this stuff, though, no self-respecting developer is going to leave it out of his software. When W3.2 comes along (or God forbid, W4!), what will be the innovations which we have to take on board then? How much working software will we have to patch up and muck about with just to keep up to date with the buzzwords? How much time will we have left for really useful research and development?

The split between IBM and Microsoft is doing more than playing out Kramer vs Kramer inside our computers. It is tearing their baby apart. All the qualities of stability, reliability, generality, and (let's face it) boredom which made the PC great are being trampled in this squalid little scrap. The principle of evolution which has stood the PC in good stead for ten years is being sacrificed as each company tries to score points over the other.

The silliest thing of all is that MS and IBM are not worlds apart. The same spirit of compromise which created the machine could rescue it now. IBM and MS still share a huge amount of technology (even if they call it by different names), and now that Apple has approached MS for help with writing its GUIs there is still going to be an almost free flow of information between the companies.

If they were my kids, I'd bang their heads together.

EXE

Jules May has never assaulted a minor (nor anyone else) and would like to make it clear that the the last sentence is a figure of speech. If you want to make something of it, you can find him on 0707 44185 or on CIX as `jules`.

Asynchronous Point-to-Point Communications or Telephone-polling Multiple Computers

XOREN COMMS SOFTWARE– Software with ten years' pedigree

Thoughout the 1980s, Xoren Computing pioneered the development of inexpensive yet reliable communications software. Today, the XOREN IPL-11 range is renowned for enabling automated file transfer between any quantity and combination of the major minis, micros and PCs in configurations which will become more and more commonplace through the 1990s.

Leading with Automated Connectivity

Although many other communications packages can asynchronously link PCs together, very few besides IPL-11 will orchestrate *fully automated connectivity* between multiple PCs and hosts.

Automated operation with IPL-11 gives users the freedom to perform unattended data communications 24 hours a day. Computers with modems operating up to 9600 baud can be scheduled to dial up remote computers over ordinary telephone lines day or night to send and receive files. Any IPL-11 version can link and transfer files to any other IPL-11 version, regardless of the computers' operating system. Once the files have arrived safely, telephone lines are disengaged.

Leading with Customer Support

Key to the success of the IPL-11 range is an emphasis by Xoren on customer support.

Because communications is notorious for being a difficult application to implement, Xoren offers installation and support services to ensure the software can operate according to the *user's* requirements.

This principle has made IPL-11 communications software the obvious choice for UK and European companies. Because IPL-11 is a British product, developed and supported in the UK, Xoren is strategically placed to support customers when they need support most - during regular business hours.

Call or write to Xoren today - the most practical solution to automated data communications has always been right on your doorstep.

XOREN IPL-11 Developments over a Decade

1980: Xoren launches IPL-11. Features include peer-to-peer file transfer under operator control, CRC error-checking on individual packets and queuing of file transfer commands.
Original versions introduced for RSX-11M and VMS for DEC PDP-11 and VAX.

1981: Remote activation facility incorporated allowing file transfers when remote computers are unmanned.
RSTS/E, RT-11 and RSX-11M PLUS versions released for DEC PDP-11.

1982: Terminal emulation facility introduced enabling the use of a terminal on a local computer as a terminal on a remote computer thereby allowing control of file transfer sessions from a single terminal.
TSX PLUS version released for DEC PDP-11.

1983: Option to control file transfers from command files as an alternative to control from operator's keyboard.
P/OS version released for DEC Professional.

1984: Mechanism built into the package protecting against "message bouncing" due to line noise when computers remain connected and the package is not in use.
MicroRSX and MicroRSTS versions released for DEC MicroPDP-11.

1985: Commenced development of new portable versions written in the programme language 'C'.

1986: First releases of new portable versions written in 'C' for PC-DOS, MS-DOS, UNIX, AIX and VMS.

1987: Portable versions support simultaneous multiple links.
IPL-11 wins ICP Million Dollar Award.

1988: PC versions enhanced with improved terminal emulation including VT100 emulation, keyboard mapping and facilities to define function keys.

1989: Comprehensive upgrade for unattended operation of multiple PC/host links supporting auto-dialling modems.

1990: **Release of MULTI-POLL range of polling software incorporating IPL-11 allowing PCs or other computers at multiple sites to be telephone-polled by a central host computer system to transfer data to and from the central system.**

XOREN COMPUTING LTD
28 MADDOX STREET
LONDON W1R 9PF
TELEPHONE 071-629 5932
FAX 071-629 5432

I am interested in Xoren Comms Software:

Name _____

Position _____

Company _____

Address _____

_____

Postcode _____

.EXE 10/91

Clever Bits

Wow your punters by displaying their company logos in text mode programs! Willie Watts presents some code to load Windows bitmaps into EGA/VGA fonts.

Like many good ideas, this is essentially somebody else's. I was working on a program to retrieve data for the .EXE Readership Survey, and looking for some sexy gimmicks to tart it up. Mr Don Binns (a name familiar to regular readers) suggested displaying a bitmapped version of the .EXE logo; he had some assembler code to do it, if I could get the image scanned into a PC file format. This seemed like a good idea. However, it took me rather longer than expected to obtain my scanned image and, what with one thing and another, I ended up doing it myself.

The Pascal program presented here displays a mono Windows .BMP file in text mode (4 and 16 colour bitmaps can't be done, as within any given character cell you can only display two colours - foreground and background). I chose the .BMP format because you can use standard utilities, such as the Paintbrush program bundled with Windows, to create and edit your bitmap files.

.BMP files

Bitmap files are reasonably (if not generously) documented in the *Microsoft Windows Programmers Reference*, and you will find structures to handle them in WINDOWS.H, the standard Windows header - the Pascal `records` in my program are translations of these C `structs`. If you pick up this reference, watch out for a set of similar OS/2 `structs` which are also included, and can confuse.

Figure 1 shows a schematic layout of a .BMP file. Since we are dealing only with mono bitmaps, most of the fields can be ignored. There are two important things to note. Don't use the `biSizeImage` field to determine the length of your bitmap before loading it - some applications seem to leave it blank. The other is that each row of bitmap data is padded to a double word boundary. Bitmap rows are ordered from bottom to top which, to anticipate myself a little, is something of an annoyance, as character maps and text screens both take *top* left as the origin. A set bit indicates a white pixel, a clear bit denotes a black.

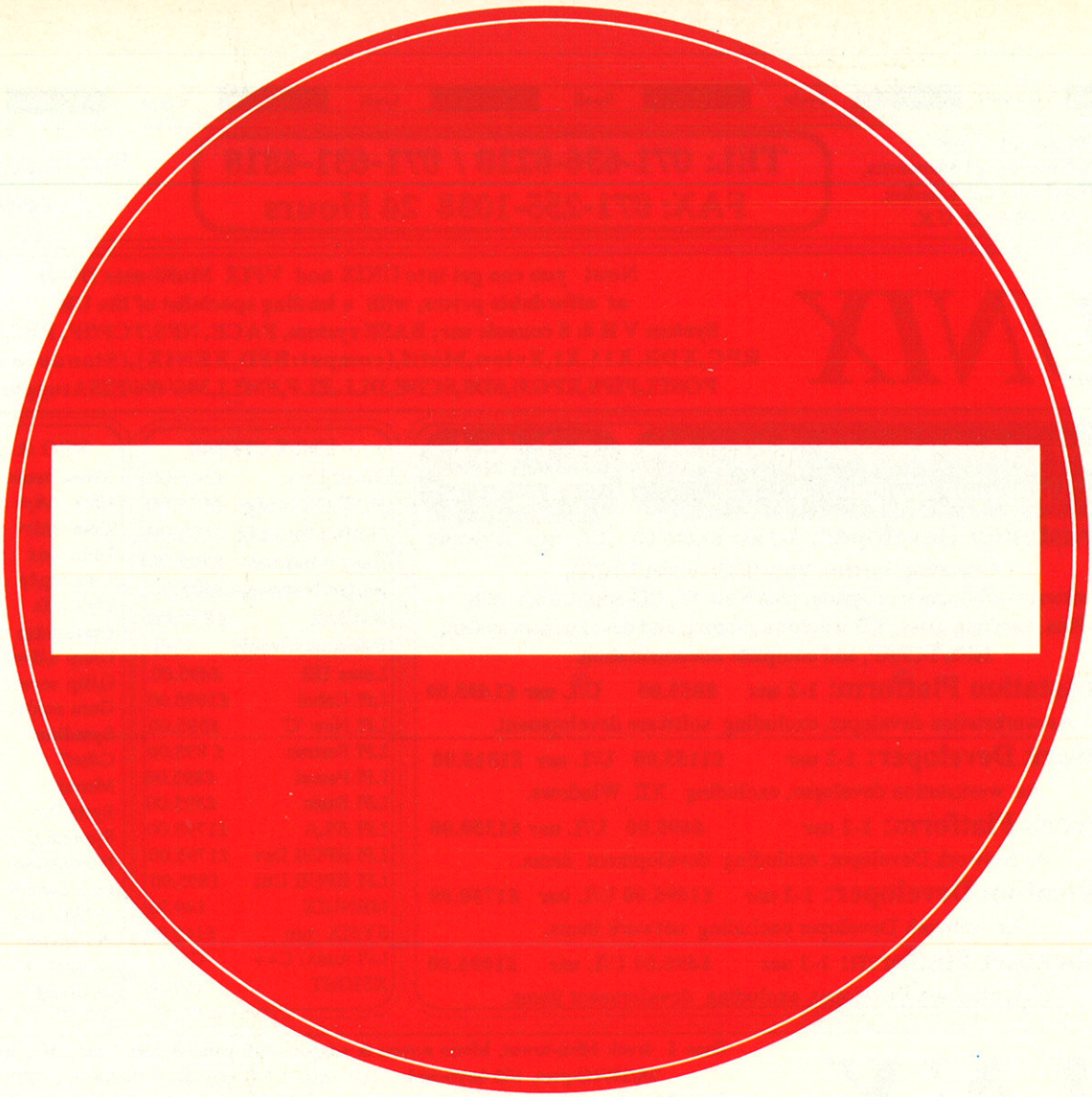
FIELD	TYPE	CONTENTS
bfType	Word	Always set to 'BM'
bfSize	Double Word	File length in bytes
bfReserved1	Word	
bfReserved2	Word	
bfOffbits	Double Word	Offset of bitmap data from start of file
biSize	Double Word	Size in bytes of BITMAPINFOHEADER, should be used to access colour info
biWidth	Double Word	Width of image in pixels
biHeight	Double Word	Height of image in pixels
biPlanes	Word	Set to 1
biBitCount	Word	Bits per pixel. =1 for mono bitmaps
biCompression	Double Word	Used with various compression constants. Mono bitmaps are not compressed, so = 0.
biSizeImage	Double Word	Number of bytes in bitmap image. Warning: some tools do not seem to fill in this field in when saving files, so I recommend that you don't use it.
biXPelsPerMeter	Double Word	Number of horizontal pixels per metre. Too ambitious for us - ignore
biYPelsPerMeter	Double Word	As above for vertical
biClrUsed	Double Word	No of colour table entries used - ignore
biClrImportant	Double Word	Most important colour table entries - ignore
|	|	|
rgbBlue[1]	Byte	Colour mappings, no use to us.
rgbGreen[1]	Byte	
rgbRed[1]	Byte	
rgbReserved[1]	Byte	
rgbBlue[2]	Byte	
etc		
|	|	|
Bitmap data	array of Double Word	Note 32-bit alignment

Figure 1 - Schematic of .BMP file

Programming Fonts

The text mode screen on a colour PC lives at address B8000H, formatted as (2000 in 80 cols x 25 rows mode) paired character:attribute bytes. Font information, which is held in the EGA/VGA card's RAM, is not usually visible within the PC's memory map. The approach used by my program is to reprogram the video controller to make the font RAM appear at address A0000H (where graphics mode video memory is usually mapped). I then write the font revisions necessary to display my bitmap. Finally I restore the memory mapping to its normal mode, and write the redefined characters to the screen in appropriate positions. I could have reprogrammed the font using BIOS Int 10H, AX = 1100H; but I should either have had to make multiple calls to the function (which causes screen flicker), or made a copy of the existing font (more code, irritating waste of 8 KB, marginally slower) - hence the macho 'let's hit the hardware' approach.

EGA/VGA fonts consist of 256 8-bit (ie 1 byte) wide by 32 deep bitmaps. Standard ROM fonts (dimensioned 8 x 8, 8 x 14 and 8 x 16) only use the top 8/14/16 bytes of each record, the remaining lines are ignored by the character generator. In fact, my program rather barbarically uses one of these lines to store a checksum. The height of a character is known as its point size; the BIOS stores the current point size at address 0040:85H.



PC Security from Microft Technology.

MENUGEN

MENUGEN provides a password protected menu interface. The system manager has complete control of what each user is permitted to do. Access to the DOS prompt can be prevented. MENUGEN has no memory overhead. **£48** + ***VAT.***

CLAM

CLAM provides TOTAL security. It incorporates MENUGEN and D-LOCK. In addition it provides a number of other facilities, the most important of which is file encryption – the only way to provide a high level of security on a micro. **£148** + ***VAT.***

D-LOCK

D-LOCK prevents access to the hard disk after booting from diskette. **£38** + ***VAT.***

S-LOCK

S-LOCK clears the screen and locks the keyboard after a specified period of inactivity or on request. The screen is restored and the keyboard unlocked only when the correct password is input. **£38** + ***VAT.***

Security Consultancy.

A professional computer security consultant can help you determine the steps required to protect your computer information and systems. The options offered range from a fixed price risk analysis with action advice to a full security audit.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 081-948 8255

CIRCLE NO. 232

Anti-virus software is also available. Corporate licences are available for all products.

Xvale Ltd., Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.

TEL: 071-636-8210 / 071-631-4818
FAX: 071-255-1038 24 Hours

Barclaycard Access

UNIX

Now! you can get into UNIX and VPIX Multi-user DOS) at affordable prices, with a leading specialist of the UK.
System V R 4: 8 console usr; BASE system, FACE, NFS,TCP/IP, RFS,DFS RPC,XDR,X11,Xt,Xview,Motif,(compat:BSD,XENIX),(Standards:SVID, POSIX,FIPS,XPG3),SDS,SCDE,DLL,ELF,FMLI,386/486/EISAsupported.

Interactive Unix SysV, REL 3.2 Architect Series: (Ver 2.21)

S.V.R.4. CALL !

Workstation Developer: 1-2 usr £1295.00 U/L usr £1995.00
Operating System, Vpix (DOS under UNIX), Software developement system plus New 'C', CO-edit, Codewatch, Ten-Plus, Looking glass, XII windows runtime and development system, NFS/TCP/IP, and complete documentation.

Workstation Platform: 1-2 usr £855.00 U/L usr £1495.00
As workstation developer, excluding software development.

Network Developer: 1-2 usr £1155.00 U/L usr £1815.00
As workstation developer, excluding XII Windows.

Network Platform: 1-2 usr £695.00 U/L usr £1350.00
As Network Developer, excluding development items.

Application Developer: 1-2 usr £1095.00 U/L usr £1750.00
As Network Developer excluding network items.

Application Platform: 1-2 usr £495.00 U/L usr £1095.00
As Application Developer, excluding development items.

UNIX ITEMS

Motif Dev	£345.00
PC-DOS bridge	£175.00
Norton for unix	£195.00
Easy windows	£595.00
Ported Netware	£3995.00
SMB/ix	£895.00
Informix/Oracle	Call
Lotus 123	£495.00
LPI Cobol	£1095.00
LPI New 'C'	£595.00
LPI Fortran	£ 895.00
LPI Pascal	£895.00
LPI Basic	£595.00
LPI /PL/1	£1795.00
LPI RPGII Dev	£1795.00
LPI RPGII Util	£895.00
MENUIX	£49.95
SYSIX -pro	£145.00
LPI (unix) C++	CALL
XSIGHT	CALL

SCO & Others

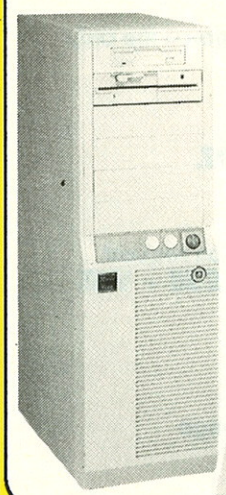
Xenix Items	Call
Unix o/sys	£525.00
Unix s/dev	£613.00
Unix vpix	£321.00
Unix tcp/ip	£227.00
Unix nfs	£340.00
Open dsktp	£795.00
O/dtp s/dev	£1095.00
O/dtp servr	£1095.00
Guru reprter	£49.95
Syseditor	£49.95
Cshell	£49.95
Menuix	£49.95
Sysix Pro	£145.00
SystemIX	£295.00
PC-Interface	£170.00
Pc-int/wndw	£225.00
TCP/IP dos	£185.00
PC-Xview	£245.00
Terminl ctl	£695.00
PC-Xsight	£395.00

LNX

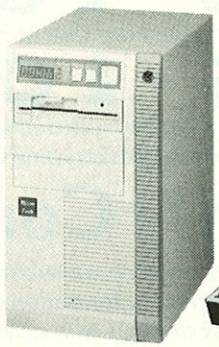
Base 1 level: Mini-tower, Mono screen & adapter with parallel port, Dual hd/fd ctrlr with 2S/1P/1G port, 102 keyboard, 1 Mb ram, 1 * floppy drive (5-1/4 or 3-1/2), Mouse.
Base 2 level: as base 1 + SVGA colour screen & 512k svga card, 4 Mb ram on m'board.
Base 3 level: as base 2 + 1 Mb ram on SVGA card, 8 Mb ram on m'board.
Base 4 level: as base 3 + 16 Mb ram on m'board, Full tower case. (e & oe)

FREE! SOFTWARE BUNDLES
a: Menuix, Unix C-shell & tools on dos
b: Unix v2.02, Vpix, Menuix.
c: Unix v2.02 Application plat,Menuix
d: Unix v2.02 Network plat,Menuix
e: Unix v2.02 Worksta plat,Menuix
f: Unix v2.02 Workstation dev,Menuix
g: Unix v2.21 Workstation dev,Menuix
h: Unix ** System V rel 4 **,Menuix

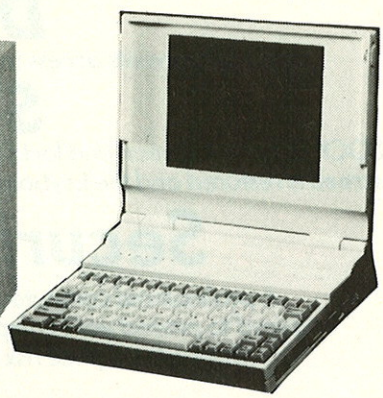
	386-20 base 1	386-25 base 2	386-33 base 2	486sx33 base 3	486-33 base 3	486-50 base 4	Unix Note book
80mb	£1040a	£1610b	£1795b	£1945c	£2795e	£3695e	
120m	£1110a	£1775b	£1860c	£2015d	£2935e	£3835e	£2495c
200m	£1185a	£1850b	£2025c	£2175d	£3050e	£3950f	£g call
600m	£1685a	£2350c	£2625d	£2775e	£3850f	£4850f	£h call



LNX systems



Unix Notebook - 20



Option A
150 MB TAPE £795

Option B
8 Port i/o with 2 * Wyse Terminal £1395

Option C
1.2 Gigbt h/d & ctrl £2495

Strategy

There's not much point in being able to load a bitmap in a text mode program if you cannot also display the text that you want. This means that we must discriminate which characters get redefined; the smiley face (code 1) is unlikely to be a great loss, lower case 'e' is more important. The constant GRABCHRS_STR, defined at the top of the program, is an array of sacrificial characters which may be redefined to build the bitmap. You can alter the contents of this array to suit your own application.

How to produce an image? My program iterates through the bitmap data in character sized cells. At each iteration, I load the relevant part of the image into a font cell (16 rows of 8 bits on a VGA machine). I copy the font cell to the next available character in GRABCHRS, and print that character in the correct position in my buffer copy of the screen. When the bitmap is complete, I simply reprogram the video hardware to its normal mode and copy my buffer back to B8000H.

I chose the .BMP format because you can use standard utilities to create and edit your bitmap files

Actually, this is a simplification. Since there are, at most, only 256 characters available for reprogramming, only very small bitmaps can be displayed. But most bitmaps contain repeated portions. If two cells display the same pattern - the most common examples being a space and a solid block - then we should reuse the same character. So, after the program has built each character cell, it checks through all those that have already been defined to see if there is a match (if I had been really keen, it would also check through the non-sacrificial characters, but I couldn't think of a way of doing this elegantly). Only if no match is found do I consume another character. This technique means that it is not possible to anticipate whether or not a bitmap can be fully displayed. My program stops when it has finished the bitmap or run out of characters.

The code

Since Pascal's handling of pointers is not as flexible as some languages *(Reader's voice: you should have used C, you silly*

```
program Bitmapld;
Uses
  DOS,
  CRT;
{FUNCTION:
  Loads and displays a mono .BMP bitmap file
  in Colour EGA/VGA text mode.
}
const
  {Width of chars in usable pixels}
  CHAR_WIDTH = 8;


{********** 'Spare' characters,
                    overwritten by bitmap}
{Edit this to suit your requirements}
  GRABCHRS_STR : String =
  #0#1#2#3#4#5#6#7#8#9#10#11#12#13#14#15 +
  #16#17#18#19#20#21#22#23#24#25#26#27#28 +
  #29#30#31 +
  '!"#$%&'#39'()*+,-./0123456789:;' +
  '<=>?@ABCDEFGHIJKLMNOPQRSTUVW' +
  'XYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~' +
  'Çüéâäàåçêëèïîìi×ÅÉæÆôöòûùÿÖÜ¢£¥ƒáíóúñÑªº' +
  '¿¡«»ß';


type
{********** .BMP file data structures}
  TBitMapFileHeader = record
    bfType          : Word;
    bfSize          : LongInt;
    bfReserved1     : Word;
    bfReserved2     : Word;
    bfOffbits       : LongInt;
  end; {TBitMapFileHeader}


  TRgbQuad          = record
    rgbBlue         : Byte;
    rgbGreen        : Byte;
    rgbRed          : Byte;
    rgbReserved     : Byte;
  end; {TRgbQuad}


  TBitMapInfoHeader = record
    biSize          : LongInt;
    biWidth         : LongInt;
    biHeight        : LongInt;
    biPlanes        : Word;
    biBitCount      : Word;
    biCompression   : LongInt;
    biSizeImage     : LongInt;
    biXPelsPerMeter : LongInt;
    biYPelsPerMeter : LongInt;
    biClrUsed       : LongInt;
    biClrImportant  : LongInt;
  end; {TBitMapInfoHeader}
  TBitMapInfo       = record
    bmHeader        : TBitMapInfoHeader;
    bmiColors       : array [0..0] of
                                TRgbQuad;
  end; {TBitMapInfo}
  TBitMapContents  = record
    case Boolean of
    True:   (bmcHeader   : TBitMapFileHeader;
             bmcInfo     : TBitMapInfo);
    False:  (bmcByte     : array[0..100]
                                of Byte);
   end; {TBitMapContents}
  PBitMapContents = ^TBitMapContents;
{********** Character Map Structures}
  TCharMap = array[0..31] of Byte;
  TCharMapArray = array[0..255] of TCharMap;


{********** Global variables}
var
  { Addresses assume colour EGA/VGA }
  CRT_COLS         : Word absolute
                                $0040:$004A;
  CRT_ROWS_MINUS_1: Byte absolute
                                $0040:$0084;
  POINTS           : Word absolute
                                $0040:$0085;
  CharMapArray     : TCharMapArray
                        absolute $A000:$0000;
  ScreenBuffer     : array[0..$fff] of Byte
                         absolute $B800:0000;
  TempVideoBuffer : array[0..$fff] of Byte;
  {Shorthand for GRABCHRS' length byte}
  NOGRABCHRS : byte absolute GRABCHRS_STR;
  {Easy access to GRABCHRS}
  GRABCHRS : array[-1..255] of Byte
                      absolute GRABCHRS_STR;
  {Stores default Exit procedure pointer}
  ExitSave : Pointer;


{********** Code begins}
function LoadBitMapFile(fname : String;
                        var FSize : LongInt)
                                : Pointer;
var
  buffer : Pointer;
  Infile : File;
begin
  Assign(Infile, fname);
{$I-}
  Reset(Infile,1);
  if IOresult <> 0 then
  begin
    Writeln('Error opening file ',fname);
    Exit;
  end;
{$I+}
  FSize := FileSize(Infile);
  GetMem(buffer,Fsize);
  Reset(Infile,Fsize);
  if buffer <> nil then
    BlockRead(Infile,buffer^,1)
  else
    WriteLn('Error: File too big to load');
  Close(Infile);
  LoadBitMapFile := Buffer;
end; {LoadBitMapFile}
procedure ProcessBitMap(Buffer :
                          PBitMapContents;
                        row, col : Byte);
{ Displays the bitmap Buffer, with top left
  corner at position (row, col)             }
type
  TByteBuffer = array[0..32767] of Byte;
  TSeqAndGCparms = array [0..3] of Word;
const
  SeqparmsSet : TSeqAndGCparms =
    ($100, $402, $704, $300);
  GCParmsSet  : TSeqAndGCparms =
    ($204, $005, $006, 0);
  SeqparmsClr : TSeqAndGCparms =
    ($100, $302, $304, $300);
  GCParmsClr  : TSeqAndGCparms =
    ($004, $1005, $0E06, 0);


var
  BMData : ^TByteBuffer;
  BMBytesPerRow,
  CurX, CurY,
  WidthInChars, HeightInChars,
  CharsUsedUp : Integer;
  CharMap : TCharMap;
  lastrow, endrow, endmask : Byte;
  procedure SetUp;
  var
    PixelWidth : Word;
    CRT_ROWS   : Byte;
  begin
    {Get start address of bitmap data}
    with Buffer^ do
      BMData :=
              @bmcByte[bmcHeader.bfOffBits];

    with Buffer^.bmcInfo.bmHeader do
    begin
      BMBytesPerRow := 4 * ((biWidth + 31)
                                  div 32);
      {Get Clipped Width}
      if biWidth > (CHAR_WIDTH * CRT_COLS)
                                      then
        biWidth := (CHAR_WIDTH * CRT_COLS);
      WidthInChars := (biWidth + CHAR_WIDTH
                        - 1) div CHAR_WIDTH;
      if biWidth =
            (WidthInChars * CHAR_WIDTH) then
        endmask := $FF
      else
        endmask := Byte($FF shl
                    (CHAR_WIDTH - biWidth
                          mod CHAR_WIDTH));
      CRT_ROWS := Succ(CRT_ROWS_MINUS_1);
      {Get Clipped Height}
      if biHeight > (POINTS * CRT_ROWS) then
        biHeight := (POINTS * CRT_ROWS);
      HeightInChars := (biHeight +
                    Pred(POINTS)) div POINTS;
      endrow := POINTS - Pred(biHeight)
                                  mod POINTS;
    end; {with}
  end; {SetUp}
```

Figure 2 - BITMAPLD.PAS listing

banana), I have resorted to a few Turbo Pascal tricks in the writing of this code. I wanted to be able to alter the characters in the GRABCHRS array without having to worry about updating a constant representing its length. To achieve this I defined a constant string `GRABCHRS_STR`, and used the `absolute` directive to position `NOGRABCHRS` over its length byte, and make `GRABCHRS` an array of bytes with element 0 at the first available character. I also use `absolute` to access various BIOS constants and screen buffers.

As I have already stated, the .BMP records are swiped from WINDOWS.H, and are explained in Figure 1. `TCharMap` and `TCharMapArray` are types used to access individual font character definitions and the complete font respectively. Of the global statics, perhaps `TempVideoBuffer` and `ExitSave` deserve individual explanation. `TempVideoBuffer` is initialised with the current contents of the text screen. As the bitmap font is built up, the program writes characters back to this buffer. When all is done, the buffer is copied back to the screen.

Turbo Pascal lets you define an Exit procedure, ie a routine that is invariably called on program termination, whether it be graceful,

I have not mentioned one big bluebottle in the otherwise clear ointment

Ctrl-Breaked or with runtime error. I define `ResetFont` as my exit procedure - this makes a BIOS call to reload the default ROM font and restore the video *status quo*, whatever idiocy has been perpetrated on the video controller. Without this setup, a program crash can leave you groping in the dark for MODE CO80. The TP manual states that you should chain to the original exit procedure, so I store a pointer to it in `ExitSave`.

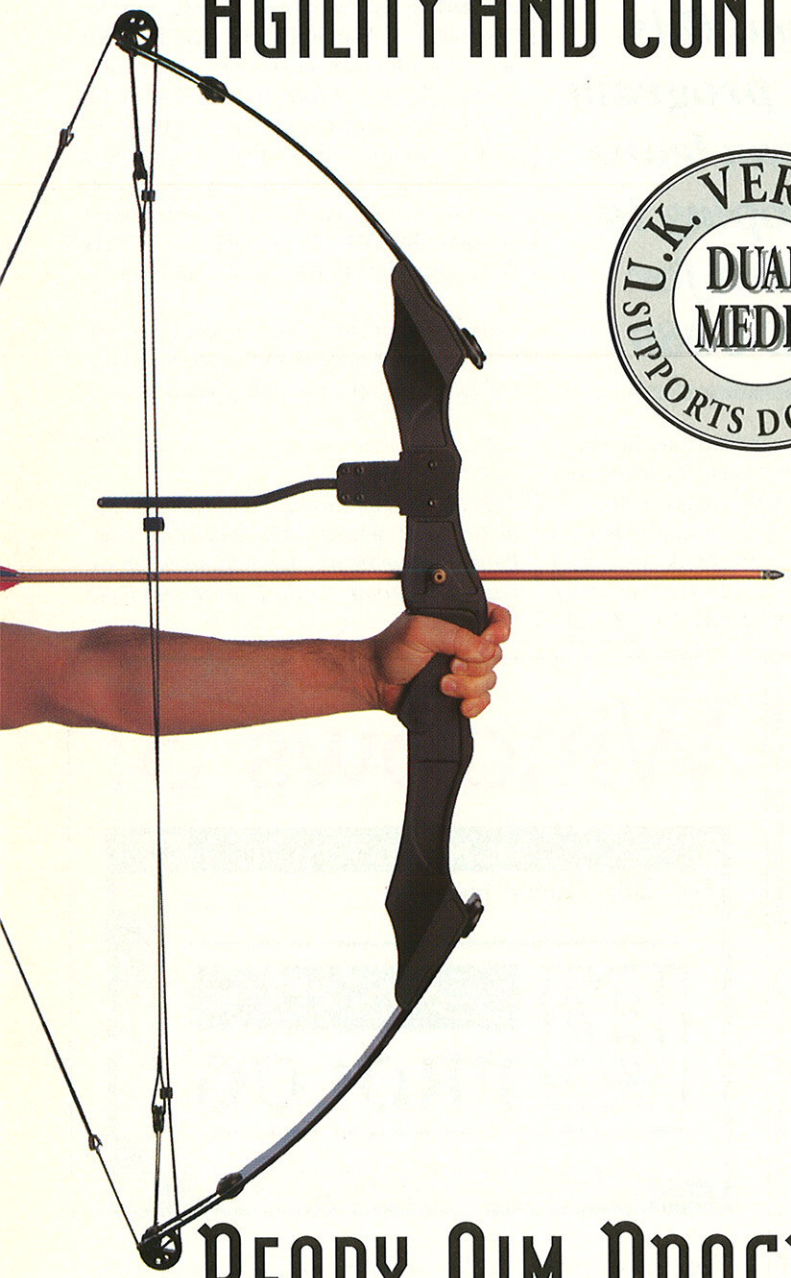
The code: `LoadBitMapFile` sucks all the data into a heap-allocated buffer. This design restricts the maximum size of file to rather less than 64 KB, but, given the maximum complexity of bitmaps we can deal with, this is not a problem.

`ProcessBitMap` does all the work. Its subprocedure `SetUp` works out some fiddly constants required to do the job, such as the number of bytes per row in the bitmap, and the AND mask required to display the right hand edge of the image correctly. `cGenMode` looks after reprogramming the video controller to get access to font RAM; it uses a set of magic numbers cribbed from Richard Wilton's brilliant *Programmer's Guide to PC & PS/2 Video Systems* (ISBN 1-55615-103-9). `MapToChar` converts one cell of the bitmap

```
procedure cGenMode(
    var Seqparms, GCparms: TSeqAndGCparms);
var i : Integer;
begin
  asm cli end; {Disable interrupts}
  for i := 0 to 3 do
    PortW[$03c4] := Seqparms[i];
  asm sti end;
  for i := 0 to 2 do
    PortW[$03ce] := GCparms[i];
end;

  procedure MapToChar;
  var
    i, offset : Integer;
    mask : Byte;
  begin
    if CurX = WidthInChars - 1 then
      mask := endmask
    else
      mask := $FF;
    offset := Succ(CurY) * POINTS *
                      BMBytesPerRow + CurX;
    CharMap[POINTS] := 0; {Zero checksum}
    for i := 0 to POINTS - 1 do
    begin
      if i >= lastrow  then
        CharMap[i] := BMData^[offset]
                                and mask
      else
        CharMap[i] := 0;
      Inc(CharMap[POINTS], CharMap[i]);
      Dec(offset,BMBytesPerRow);
    end;
  end; {MapToChar}

  procedure PlotChar(character : Byte);
  begin
    TempVideoBuffer[((HeightInChars - CurY +
                    row - 1) * CRT_COLS +
                          CurX + col) * 2]
                               := character;
  end; {PlotChar}

  function FoundInStore : Boolean;
    function Comp(var Buf1, Buf2 : TCharMap;
                      POINTS     : Word)
                        : Boolean; assembler;
    asm
      push      ds
{ Move default return to ax - 'True'}
      mov       ax, 1
      lds       si, Buf1
      add       si, [POINTS]
      les       di, Buf2
      add       di, [POINTS]
      mov       cx, [POINTS]
      inc       cx
{ Search backwards to find checksum 1st}
      std
      rep       cmpsb
      je        @Exit
{ Flag failed match in return }
      xor       ax, ax
@Exit:
      pop       ds
end;
  var
    i : Integer;
  begin
    FoundInStore := True;
    for i := 0 to Pred(CharsUsedUp) do
      if Comp(CharMapArray[GRABCHRS[i]],
              CharMap,POINTS)          then
      begin
        PlotChar(GRABCHRS[i]);
        Exit;
      end;
    FoundInStore := False;
  end;

begin  { ProcessBitMap }
  Setup;
  Move(ScreenBuffer, TempVideoBuffer,
                  Sizeof(TempVideoBuffer));
  cGenMode(SeqparmsSet, GCParmsSet);
  CharsUsedUp := 0; lastrow := 0;
  CurX := 0; CurY := 0;
  repeat
    MapToChar;
    if not FoundInStore then
    begin
      Move(CharMap,
           CharMapArray[
                  GRABCHRS[CharsUsedUp]],
           Succ(Points));
      PlotChar(GRABCHRS[CharsUsedUp]);
      Inc(CharsUsedUp);
    end;
    CurX := (CurX + 1) mod WidthInChars;
    if (CurX = 0) then
    begin
      Inc(CurY);
      if CurY = HeightInChars - 1 then
        lastrow := endrow
    end;
  until (CharsUsedUp = NOGRABCHRS) or
        (CurY = HeightInChars);
  cGenMode(SeqparmsClr, GCParmsClr);
  Move(TempVideoBuffer, ScreenBuffer,
       Sizeof(TempVideoBuffer));
end; {ProcessBitMap}

procedure DisplayBitMap;
var
  BitMapBuffer : PBitMapContents;
  FileSize : LongInt;
begin
  if ParamStr(1) = '' then
  begin
    WriteLn('USAGE: BITMAPLD <file.bmp>');
    Exit;
  end;
  BitMapBuffer :=
       LoadBitMapFile(ParamStr(1),FileSize);
  if BitMapBuffer<> nil then
  begin
    ProcessBitMap(BitMapBuffer,0,0);
    FreeMem(BitMapBuffer, FileSize);
  end;
end; {DisplayBitMap}


procedure ResetFont; far;
{This is called as an exit procedure, so the
 screen is always restored, even if something
 goes wrong... not that it will...   }
var
  regs : Registers;
begin
  {Restore default exit proc}
  ExitProc := ExitSave;
  regs.ah := $11;
  case POINTS of {Select correct ROM font}
    16: {Load 8 x 16 font}
      regs.al := 4;
    8:  {Load 8 x 8 font}
      regs.al := 2;
    else {Choose 8 x 14 - it's the safest}
      regs.al := 1;
  end; {case}
  regs.bl := 0;
  intr($10,regs);
end; {ResetFont}


{********** Main program }
begin
  ExitSave := ExitProc;
  ExitProc := @ResetFont;
  ClrScr;
  DisplayBitMap;
  ReadKey;
end.
```

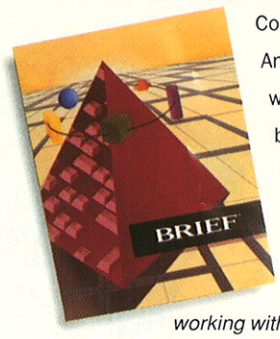
Figure 2 - BITMAPLD.PAS listing (continued)

BRIEF'S NEW MOUSE SUPPORT GIVES YOU GREATER AGILITY AND CONTROL.



U.K. VERSION
DUAL MEDIA
SUPPORTS DOS & OS/2

More than 100,000 programmers around the world swear by the virtues of BRIEF – its unmatched power, flexibility, and intuitive command structure. Now, thanks to mouse support and other new features, version 3.1 of the best-selling text editor will make you quicker and more productive than ever. ¶ Hit a button on the mouse – a special, user-configurable menu window pops up and lets you execute any command you wish. Mark text and move around in your windows without using the keyboard. Zoom your windows to fill the screen. ¶ As snappy as it is, the mouse is only one enhancement among many. BRIEF 3.1 also takes advantage of EMS memory whenever it reads in a file. The result: faster performance, whether you're editing a file or running a macro. ¶ So what else is new? Redo – a step for step complement to undo. Support for Microsoft C 6.0 Advisor and Microsoft's Programmer's Workbench. And more. ¶ BRIEF's seamless interface with other Solution Systems products gives you instant access to the tools you need. A single keystroke lets you keep track of product development with Sourcerer's Apprentice – Version Control for the Professional. Analyze product performance with CHARGE. And nothing beats dBRIEF for tailoring BRIEF to Dbase or Paradox. ¶ *BRIEF sells for £199 excluding VAT.* ¶ *Once you start working with BRIEF 3.1 you'll really fly.*



READY, AIM, PROGRAM.

Order your copy of BRIEF today.

call:

0763 244141

Solutionsystems
THE PHYSICS OF PROGRAMMING

1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763-244141
Facsimile: 0763-244025

CIRCLE NO. 234

image into font definition format. `PlotChar` moves a character to the correct position in `TempVideoBuffer`. Function `FoundInStore` searches for a match between the current cell and those already defined. This is the most time-consuming part of the program, so I broke out in a rash of inline assembler language. I store a checksum at the bottom of each character matrix, to encourage compares to end ASAP. The action of `ProcessBitMap` is fairly obvious, although I note for the benefit of non-Pascalians that `Move` is equivalent to ANSI C's `memmove()`.

Most of the remainder of the program is simple housekeeping. `ResetFont`, which restores the ROM font, is, like the rest of the program, written to be independent of the text mode in which the EGA/VGA card is running (although I do lazily assume a colour screen).

End thoughts

I have not mentioned one big bluebottle in the otherwise clear ointment. VGA standard colour text mode is 720 pixels wide by 400 deep. This implies that each character should be nine bits wide, yet I have stated that they are only eight. The answer is that the font bits are used to fill the first eight bits of the width, and the bit on the right hand side of each character is left blank. Loaded bitmaps appear with parallel vertical lines. There is worse. In order to get box drawing characters to join up, there is a fudge by which, for characters C0H to DFH, the value of the rightmost bit is copied into the ninth. If you use these as sacrificial characters, your display will have vertical streaks with acne. Probably the easiest way around this is to force VGAs into 640 x 350 mode, by calling Int 10H, AX = 1201H (or 1202H to get back), BL = 30H followed by Int 10H, AX = 0003H.

Without this setup, a program crash can leave you groping in the dark for MODE CO80

If you try to load a large, complicated bitmap, such as Window's PAPER.BMP, you will run out of characters. You can extend the range of this program by running the display adapter in 512 character set mode. Both EGAs and VGAs support this feature; a call to Int 10H, AX = 1103H, BL = 4 (BL = 0 restores normality) will set you up to use a second 256 character font starting at A4000H as mapped into CPU memory by `cGenMode`. The problem is that, in order select the other 256 characters, the adapter borrows bit 3 - the foreground intensity bit - of the accompanying attribute byte. So you must either accept that all your new characters will be displayed in different colours, or disable that bit of the foreground attribute (Int 10H, AX = 1000H, BX = 0712H does this, call with BX = 0F12H to reset).

I would like to have shown you the fruits of my labours, but none of my text mode screen-grabbers can cope with remapped fonts...

EXE

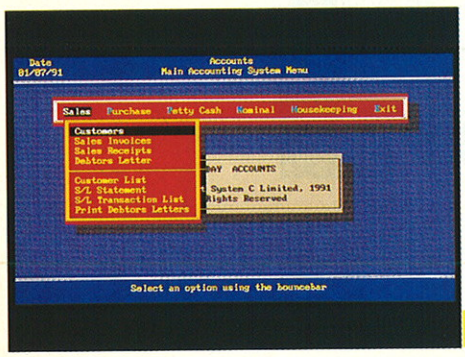
The code accompanying this article is available on disk. Send a blank floppy disk to the Editor, following the instructions given on Page 1, column 1. Mark your envelopes 'BITS'.

Spend two free hours with Sycero dB and save days of development time

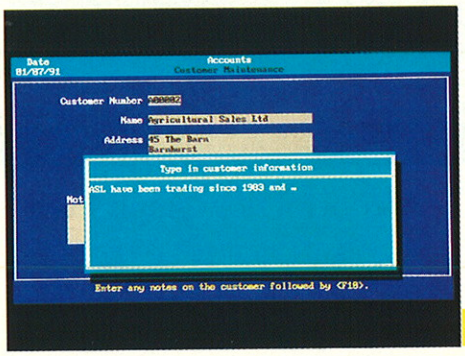
Imagine being able to develop applications quickly, simply and reliably — with the minimum of debugging. And in a fraction of the time. But without losing any of the flexibility.

Now you can. With Sycero dB, a complete and powerful program generator that lets you develop applications in a fraction of the time it takes to do manually — by automating most of your programming. What's more, Sycero dB actually lets you add code and subroutines *before* generation. So you have the same flexibility that you value with manual coding.

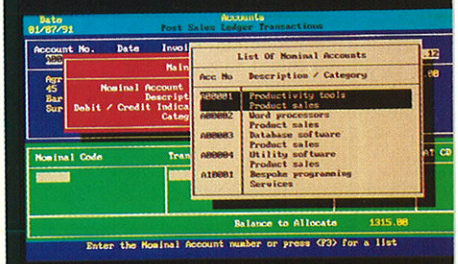
Clipper is the applications development standard for PC database management systems. Its open architecture and superior networking capabilities let you create really sophisticated applications.



1



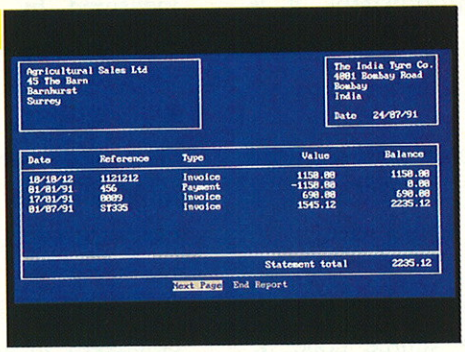
2



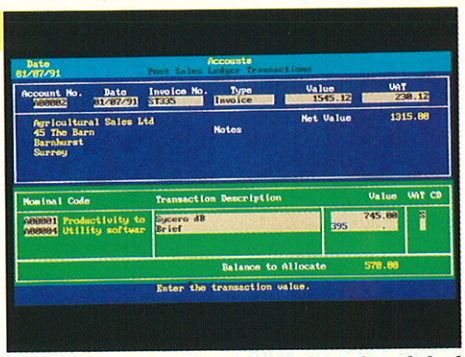
3

1. *Create pull-down, bounce-bar and Novell-style menus in minutes.*
2. *Use memo fields to give you on-screen word processing in a window.*
3. *Overlay a browse window with a function key in just five lines of code.*
4. *A complete and powerful report generator for forms, lists and more complex invoices/statements.*
5. *Transactions or multiple records per screen produced easily.*

4



5



All trademarks are acknowledged.

Sycero makes use of Clipper's advanced facilities, and goes a stage further. Sycero dB NET actually builds in the automatic file and record locking required by network systems.

Together, Sycero and Clipper form the perfect development environment. But don't just take our word for it. Seeing is believing, and time is money. So spend just two hours with System C at one of our free Sycero dB seminars.

By building a suite of programs from scratch, we'll demonstrate how you can use Sycero to utilise the same development tools for both simple and complex systems; ensure complete programming standards; take the work out of networking, *and* save days of development time.

We know you'll be impressed with what you can do with Sycero in just a couple of hours. So think how productive you could be if you were using it all the time.

To find out more, simply complete and return the coupon, or contact System C on 0622 691616. It's your entry point to extra power and productivity.

nantucket®

CIRCLE NO. 237

YES. I want to know how to create more powerful and sophisticated applications.
Send me more information about ❑ Sycero dB ❑ your next Seminar ❑ the System C product range

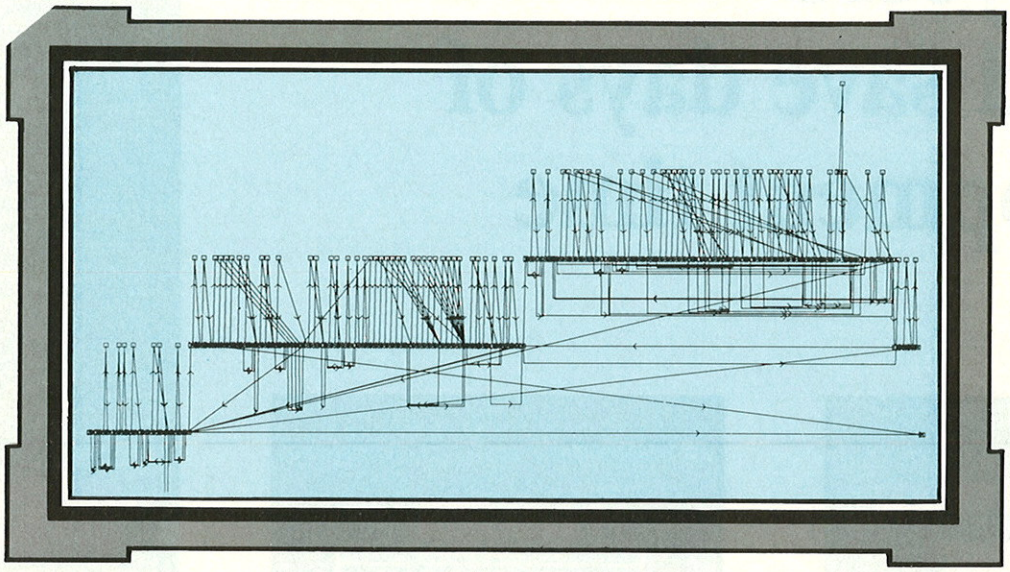
Name ..... Position .....
Company .....
Address .....
.....
Post Code ..... Tel No .....

.EXE 10/91

I am currently using the following products:
.....
.....

Please return to: System C Ltd., 60/61 High Street, Maidstone, Kent ME14 1SR. Tel: 0622 691616 Fax: 0622 691241.

ART or RE-ENGINEERING?



A control flow graph from

LOGISCOPE

◆ This could be an example of post-modernist art by a famous painter. Unfortunately, it is LOGISCOPE'S graphics analysis of some very poor source code.

◆ To help you get the picture on your software quality VERILOG developed LOGISCOPE, a powerful software analysis and testing tool.

◆ LOGISCOPE provides you with a concise view of the structure of your source code. It allows you to see a picture of the decision paths that run through each module and by using VERILOG's Call Graph a view of how they are connected is easily obtained.

◆ The quality of your code can be analysed and measured by using a comprehensive list of software metrics generated by LOGISCOPE.

◆ LOGISCOPE can help you clearly define your test programs and provide invaluable information on their effectiveness with coverage rate results and analysis.

◆ LOGISCOPE is available for most common languages including: Pascal, C, Cobol, Fortran, Modula 2, PLM, Assembler and Ada.

VERILOG's team of re-engineering professionals are ready to help you implement a case solution to solve the toughest of re-engineering challenges.

For further information about LOGISCOPE or any other VERILOG case tools, please write to Bernadette GLOVER, VERILOG UK Ltd, 34 The Quadrant, Richmond, SURREY TW9 1DN or ring us on 081 940 2212, or fax us on 081 940 2933.

VERILOG

CIRCLE NO. 238

Tools for program development

UNIX debuggers have come on a long way in recent times. Peter Collinson compares and contrasts a commercial offering and a home-grown PD program.

The X window system is making UNIX based graphical tools more available. However, it does seem to be taking a long time for UNIX to develop the level of graphical tools that are prevalent in the PC world. For example, when I came across CodeView from Microsoft, I realised that UNIX did not have anything as powerful for program development. Time has passed and there are now two UNIX based tools that, I think, are better than CodeView.

Both provide program development aids for the workstation user. The first, Saber-C, is a fully commercially supported product that provides what can best be described as a C program development workbench. The second, `ups` is public domain software originating at the University of Kent, UK. It is a program debugging tool aimed at replacing the old terminal driven `dbx` or `adb` by a graphical direct manipulation interface.

Saber-C

The engine of Saber-C is a C interpreter. The key idea is that you take your C program, consisting of several source files, and load them into Saber-C's interpreter. Loading the program is accompanied by extensive checks on the code. Each module in the program is linked dynamically with all the others and relevant libraries to generate a runnable program. Saber-C refers to all the files that comprise a runnable program as a *project*.

Once compiled and linked, the program can be run. Run-time checks can be made because the C is interpreted. For instance, Saber-C carries out array bound checking and ensures that pointers contain valid addresses.

Interpreting code for a huge application would be slow. To gain efficiency, Saber-C can deal with fully compiled code modules. A file can be loaded into Saber-C either in source form or as an object code module. The `-g` option to the compiler inserts debugging information into the object module. When the file is loaded as an object, this information is used to allow you to set breakpoints on source lines and examine variables. All you lose is the run-time checking of the interpreted source.

Figure 1 shows the Saber-C display that appears on my screen. I use a research window manager called `twm`, so don't be confused that the borders around the windows are different from yours, that's just a feature of X. I am debugging a little X application of mine called `xcal`, you can see the output from the application in the top window of the figure. Yes, it's a calendar.

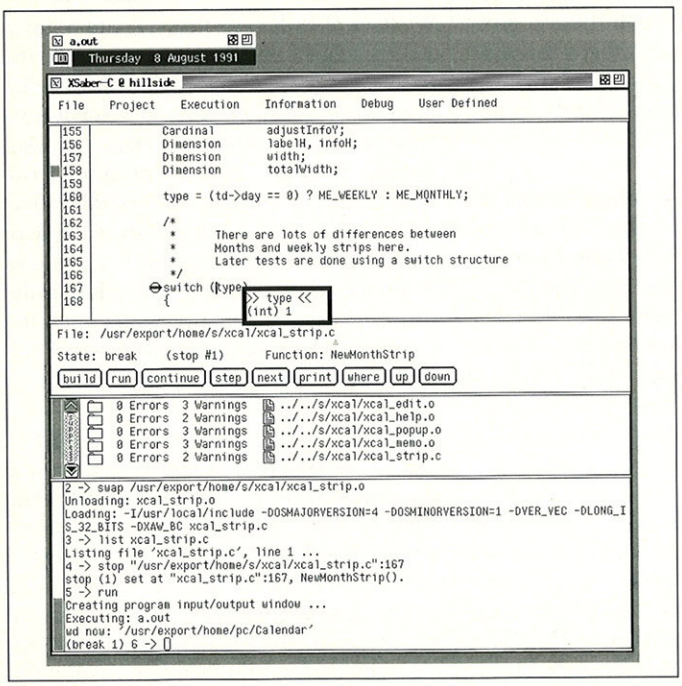


Figure 1 - Using Saber-C to debug xcal, an X application

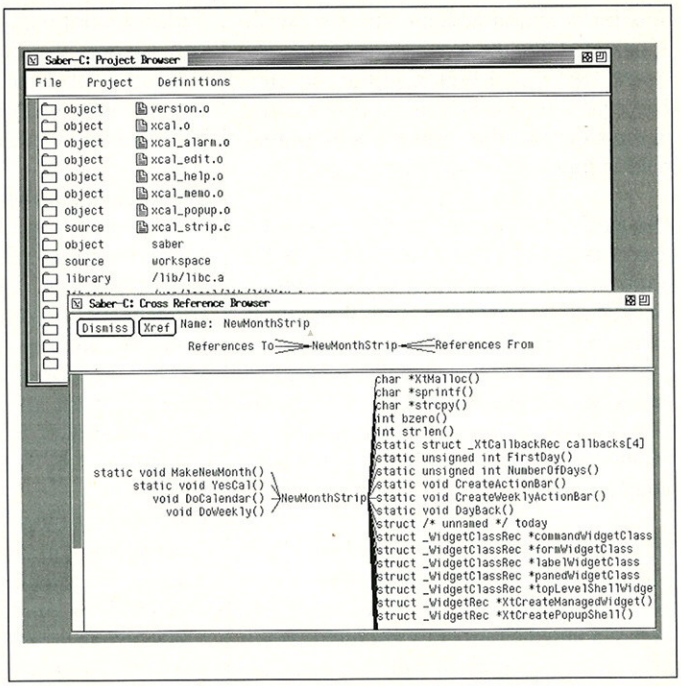


Figure 2 - Two Saber-C browsers in use on xcal

The big window below is Saber-C's main control area. I have set a breakpoint in the file `xcal_strip.c` at line 167. I triggered the breakpoint by selecting the main date area of the application. The program has paused. By pointing the mouse at `type` (on line 167) and pressing Shift on the keyboard along with the right mouse button I have obtained the contents of the `type` variable in a small transient box. The box disappears when I move the mouse.

The Saber-C display

The Saber-C display consists of several sections. The bottom segment of this window is the 'workspace'. Commands can be typed in here to make Saber-C take actions for you. Many of the buttons like `build` or `run` will also echo a command in the workspace since the buttons are regarded as short-cuts for typed commands. You can see this in the example. The `run` command was actually invoked by poking at the `run` button with the mouse. Users of Saber-C on ASCII-only terminals see just this command stream.

The area above the workspace is used to display any errors that may occur. There are two types of errors that are displayed here: compile-time (or load) errors and run-time errors. I have left some warnings in place so that you can see what they are like. Saber-C provides sophisticated control of errors so that you can suppress warnings and only see the errors that are relevant to you.

The area above that contains a set of buttons for common actions. The set can be augmented by the user by selecting the `User Defined` button in the top area. This central button area also contains status. In the example, the program is in a break state in function `NewMonthStrip()`.

The next area up the screen contains a source listing of a file. The appropriate file pops up here if you 'open' an error report by clicking on the folder symbol next to it. In the example the file `xcal_strip.c` is being displayed because I asked for it to be 'listed', see line three of the workspace. I then scrolled down the file and set a breakpoint by simply pointing at the relevant line. All very easy.

Saber-C has several tools called browsers that are used to convey information about the job in hand. I hate the word 'browser', I keep having visions of fly encrusted cows waving their tails from side to side. The Saber-C browsers are perhaps a little more awake. They pop-up in separate windows that can be placed where you want. I show two in Figure 2. The bottom window shows the Cross Reference Browser allowing you to see where a particular routine fits into the program, who calls the routine and what code is called by the routine.

I hate the word 'browser'. I keep having visions of fly-encrusted cows waving their tails from side to side

Projects

The topmost window in Figure 2 shows the Project Browser displaying all the files that are involved in the program. You can see that most of the files are loaded as compiled modules. The `xcal_strip.c` file is loaded as source. The project also contains the libraries that are used by the program. You can see the standard C library, the other libraries are the standard X set.

Clicking on the folder symbol with the left button will display the contents of the object, the names of the routines and global variables are displayed. Clicking on the folder symbol with the right button gives you a menu of options. Perhaps the most interesting of these is `swap`, this allows you to replace a compiled object module by source or vice versa.

If you click on the the other symbol attached to each module, the folded bit of paper, then you are placed into an editor so that you can edit the file. The actual editor that is used is taken from the EDITOR environment variable so it adapts to use *your* editor. This is a nice touch. It means that `vi` users can deal with files using keystrokes that they know and understand.

The notion of the project is a strong one in Saber-C. You can save a project in a text file. Later on, you can restore the state of Saber-C exactly how you had left it. It remembers which files are loaded as compiled modules and which as source files. It stores the breakpoints and what errors have been suppressed, and it also stores the setting of the various option strings.

Loading an existing project into Saber-C is a bit of a struggle at first. The key is to use `make` to do the work for you. If you add the following target into your `makefile`

```
saber_obj:
 #setopt load_flags $(CFLAGS) \
              $(INCLUDE_LIST)
 #use $(DIR_LIST)
 #load $(OBJS) $(LIBS)
 #link
```

and then load Saber-C. Typing

```
make saber_obj
```

into Saber-C will then execute the Saber-C commands. The lines that start with '#' are comments to `make` but are picked up by Saber-C as commands to be run.

This will set the flags that are to be used to load the project and set the directories to be used to find some of the source. The `load` command will load all the compiled modules into Saber-C and finally the `link` command will link all the modules with the libraries to make a runnable program.

I have one gripe with all this, most versions of `make` allow you to have `Makefile` and `makefile` in the same directory. The lower-cased name is used before the other. I tend to use this feature for testing, I will work using `Makefile` but when I need a quick test I will create `makefile`. Saber-C doesn't like this at all, and won't countenance a file called `makefile` or `Makefile` in the same directory.

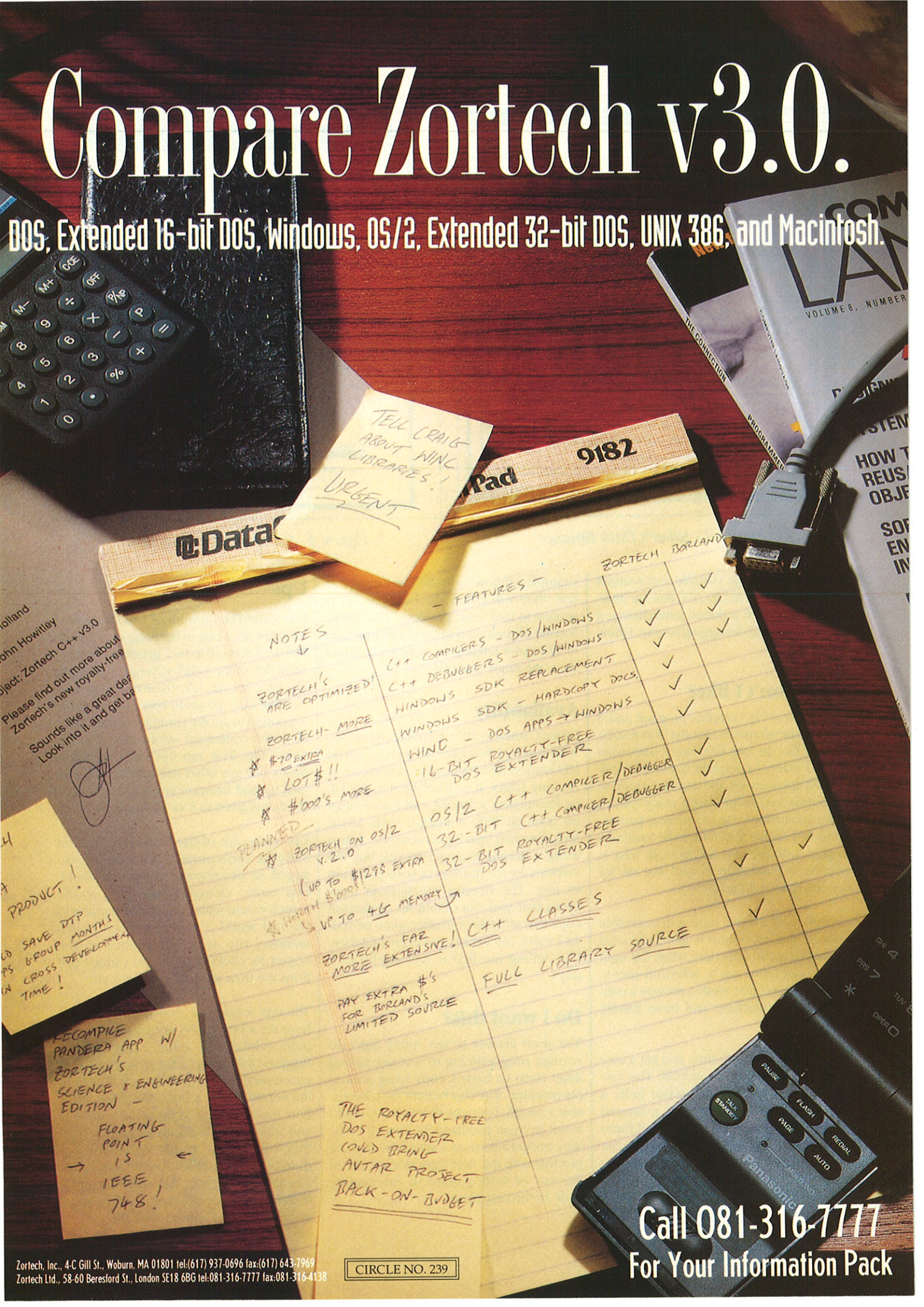
Viewing data/debugging

We have already seen that you can point at an object like a variable or a structure in the source listing area and get the current value popped up on the screen in a small box. Saber-C also provides a Data Browser where program objects ranging from simple integer variables to complex data structures can be examined. An example of this is shown in Figure 3, this is a demo of X11's widget structure. If the value in the variable changes in the program, then the displayed value in the Data Browser will track this and show the updated value.

Each object is shown in a separate box and this means different instances of the same structure can be displayed easily. The Data Browser also understands about pointers. If you follow pointers using the menu then the boxes displaying each element are linked by lines giving a graphical representation of the structure. This data display stuff seems ok, even flashy. I do feel that I would appreciate the ability to look at the same bit pattern and interpret it in different ways. This inability is a failing of many debuggers.

Compare Zortech v3.0.

DOS, Extended 16-bit DOS, Windows, OS/2, Extended 32-bit DOS, UNIX 386, and Macintosh.



Call 081-316-7777
For Your Information Pack

Zortech, Inc., 4-C Gill St., Woburn, MA 01801 tel:(617) 937-0696 fax:(617) 643-7969
Zortech Ltd., 58-60 Beresford St., London SE18 6BG tel:081-316-7777 fax:081-316-4138

CIRCLE NO. 239

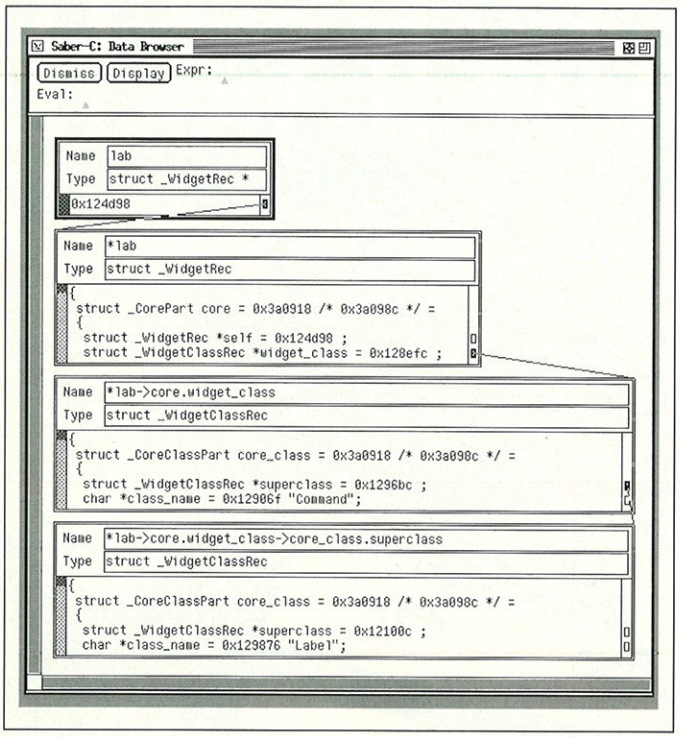


Figure 3 - Saber's Data Browser

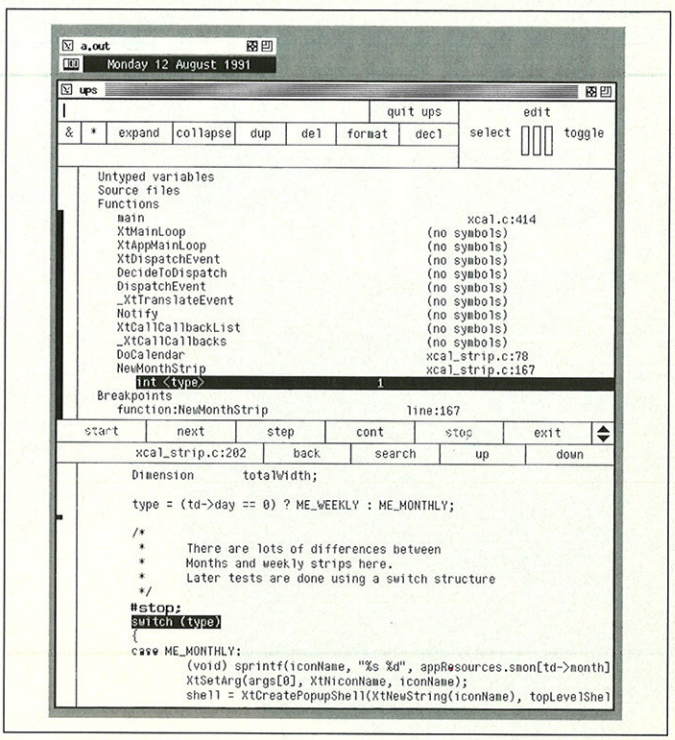


Figure 4 - Using ups to debug xcal

You can look at any variable, run routines stand-alone, type in bits of C that are interpreted and so are actioned immediately; it should be a good debugging system. I think that it is.

What don't I like?

Saber-C is a very complex program and it needs perseverance to get the best results. I do feel that the user interface is somewhat hard to learn. I complain about this with many X applications, the interfaces start off as complicated entities with many hidden key combinations used to achieve particular ends. I dislike this. I want the interfaces to grow with the user, to highlight what is possible in painful detail until you ask for the descriptions to go away.

Saber-C does use the mouse in a sensible fashion, the left button is generally used for pointing and selection; while the right button summons up a selection menu. The designers have tried to present a consistent user interface and have mostly succeeded.

The documentation is fine, you get a user guide and a reference manual. The system comes with tons of on-line help; a tour through the system and a tutorial guide. I think that it's a pain that all these documents and scripts use exactly the same example program demonstrating exactly the same bug. I expect that this was a conscious design decision, but it fails. Users need all sorts of different examples of how to do different things. At first, they will follow the example that is nearest to their own problem. This means that the documentation should include a set of examples ranging in complexity rather than one single example.

Availability

Saber-C runs on all Sun architectures running SunOS 3.5 or later. It can also run on DECstation and VAXstation workstations running Ultrix. It is distributed in the UK by Pacemaker Software, call Paul Frogatt on 0666 825855. A license costs £2400 including first year's maintenance. It runs using a license manager so you buy the software by the number of seats you want to use. The license mechanism does mean that you can obtain an evaluation copy that will time-out after some period. As I write, my evaluation copy of Saber-C is telling me that it will expire in 5 days.

Do I want this?

The short answer is 'yes', but I don't do enough programming to warrant paying large sums to have a copy lying around my machine. The cost benefit isn't favourable enough. I can see that Saber-C could result in huge productivity leaps by full-time programmers as long as the 'fun' of playing with the debugger doesn't get in the way of writing correctly functioning code in the first place.

Ups

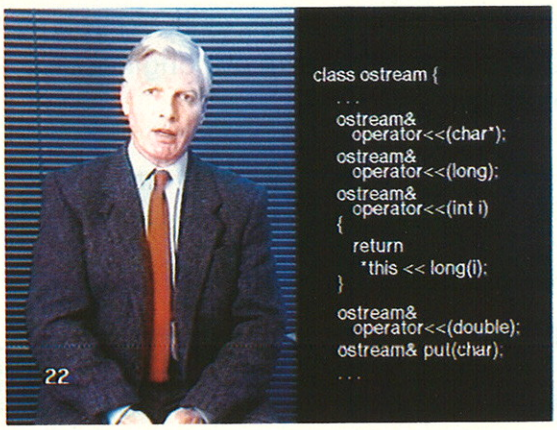
Ups should be pronounced 'Oops', as in 'I have just got a bug in my program. How do I fix it?' It sets out to be a graphical replacement for other UNIX debugging systems like `dbx` or `cdb`. Like these tools you must compile your program specially using the `-g` option to get the compiler to pass loads of useful contextual information into the object file. You run ups on a compiled binary like any other debugger. It will also pick up and interpret any data in a `core` file that is given to it as an argument.

Ups came out of the University of Kent Software Tools project. The team was engaged in developing 'direct manipulation' tools - changing something on the screen would make something happen in the real world. Ups is controlled in this manner; you select something on the screen by clicking on it and various options are then made available to you.

The ups window is shown in Figure 4. I have set a breakpoint at exactly the same point as I did in the Saber-C example, and triggered it similarly. The ups screen is split into several rectangular areas. At the top of the window on the left is the *typing line*. This is shown by a vertical bar cursor. If you type characters into the ups screen they will appear in this region and are used by several different commands, for example the `search` button in the centre of the window uses this text as an argument to search for text in the source file.

VIDEO COURSE

Learn C++



NO RUSH!
NO TRAVEL!
NO HOTELS!
ONLY £299.95

Learn C++ Now!

The great exodus of programmers from C to C++ has begun! Since C++ builds on C, it's the easiest OOP language to learn. That's why it's called "the language of the 90's".

Why the rush?

PRODUCTIVITY! Yes, C++ programmers can write programs in less time requiring less maintenance. Large projects become much easier to manage.

Unfortunately, learning C++ can be very costly. Classroom instruction is expensive even without the travel and hotel costs. Of course, not learning will cost you even more in the long run. Now there is an alternative!

The top C++ video tutorial at the lowest possible price.

The C++ video tutorial from Zortech is the ultimate C++ training tool for work or home at only £299.95. It comes on six VHS video tapes containing 32 clear, extensive tutorials.

Used in conjunction with the concise workbook and tutorial disk, you will find everything you need for fast-track C++ tuition.

The course is generic (i.e. compiler and hardware independent) and is available with or without the award winning Zortech C++ Compiler for MS-DOS and OS/2.

Start writing C++ code within a week.

As a C programmer, you will start producing C++ code within a week of concentrated use of this course. Alternatively, spend just an hour a day watching the video and working through the suggested exercises to learn C++ in only six weeks!

The Leader in video tutorials.

If you don't already know C, you can join tens of thousands of programmers world-wide who have learnt C with the Zortech Complete C Video Course, described as:

"An excellent bargain ...
... I heartily recommend"
Gary Ray, PC WEEK

Save your company thousands of £££'s.

One programmer can train for only £299.95, but you can train ten programmers for just:

C++ Course	£299.95
9 Extra Workbooks	£179.55
Total 10 students	£479.50

Yes! Only £47.95 each! (With all the FREE refresher courses you need!)

Now, Zortech with its new C++ Video Tutorial has refined the art of video tuition and presentation even further for the 1990's.

Only £299.95 complete

- *Six Videos with 32 lessons*
- *256 page workbook*
- *Tutorial disk*
- *Compiler & hardware independent*
- *NTSC or PAL format*
- *Tax deductible*

Don't delay, order now!

Just mail the coupon or call the order hotline for same day shipment.

USA: Zortech Inc.,
1165 Massachusetts Avenue,
ARLINGTON, MA02174
Voice: 617-646-6703
Fax: 617-643-7969

EUROPE: Zortech Ltd.,
106-108 Powis Street, LONDON
SE18 6LU
Voice: 44- 1-316-7777
Fax: 44- 1-316-4138

IN EUROPE CALL:

44- 1-316-7777

ORDER FORM

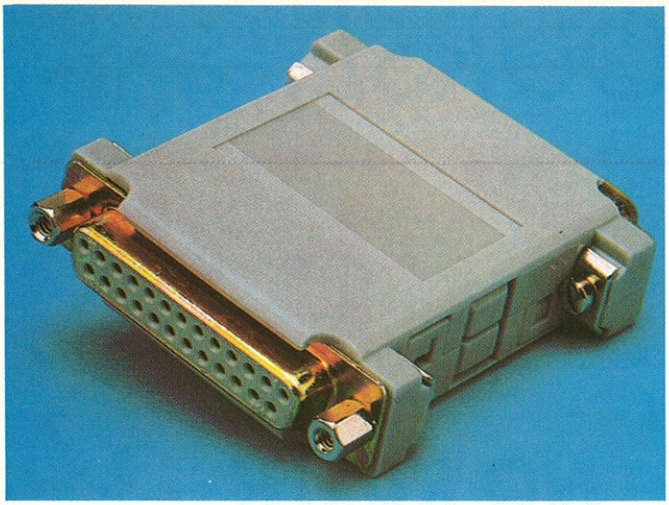
Please rush me these items:

Qty	Description	Price
___	C++ Video Tutorial	£299.95
___	Extra C++ Workbooks	£. 19.95
___	Zortech C++ Compiler	£129.95
___	Developer's Edition	£299.95

Please add £5 + VAT postage in UK
All UK orders please add 15% VAT
International shipping charged at cost

Name _____
Company _____
Address _____
_____
_____ .EXE 10/91
Phone _____
MC,VISA or CHQ_____ Expiry ____
Card No. _____

CIRCLE NO. 240



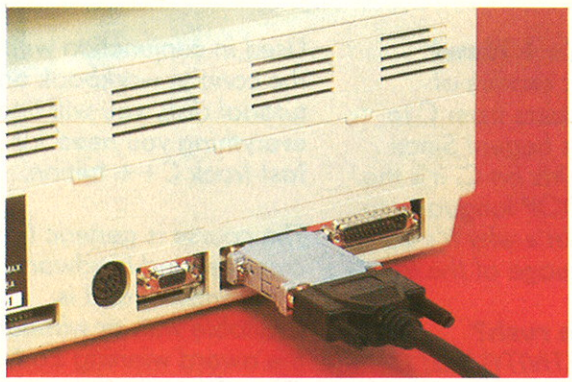
How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.



As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

- For IBM PC, PS/2 and compatibles
- Uses parallel printer port
- Totally transparent to printer
- Secure data & password can be changed from your application software
- Cascadeable
- 240 bytes of secure read/write memory
- 8 byte (64 bit) password
- No programming adaptors required
- Easy to use SOFTLoK II setup software
- Routines ready to link with various compilers
- Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

CIRCLE NO. 241

To the right of this line, is the `quit ups` button. Pressing this will generate another sub-menu asking for confirmation of your desire to leave. To the right of this is the *mousehole.* This has a representation of the three mouse buttons with a caption saying what each will do. The captions alter as you move from region to region on the screen, this provides a novice with some idea of what they can do at any point. Ups uses the mouse buttons in a consistent way. The left button is always used for selection. The centre button enables editing of some object on the screen. The right button changes its function depending on the area of screen under the cursor.

Dynamic menus

Underneath the typing line is the *dynamic menu area.* This contains a set of buttons that change depending on what object has been selected. In the example, the highlighted line `int <type>` has been selected in the display area. This is a variable in the program and the dynamic menu contains options that can be applied to variables. Hitting various buttons in the menu will affect the way that the data is shown in the display area.

The ampersand and star operators allow you to manipulate pointers and their contents. The two buttons marked `expand` and `collapse` operate on structures, showing their contents in full gory details or reclaiming the screen real-estate. The `dup` button duplicates the item, so you can show the same object several times in different formats. The `del` button deletes the object from the display area. The `format` button allows you to change the the way that an object is displayed. A sub-menu permits you to look at a variable as decimal, octal, hex, binary, ASCII characters or a character string. Finally, the `decl` button affects the treatment of the display `typedefed` objects.

The display area

The display area is where most of the interaction with the user is done. This is scrollable by placing the mouse into the scroll area on the left. Scrolling in ups is mouse sensitive; you can control the speed and direction of the scroll by holding the left mouse button down and moving it. Moving further away from the point where you pressed the button makes the scroll action go faster. The direction of scroll is determined by the movement of the mouse too. I like this, it means that you manage exactly what you see in a very natural manner.

At start-up time, the display area contains a number of buttons. You can't see all of these in the example, some of them are scrolled up. You can see `Untyped variables`, `Source files`, etc.

Saber-C could result in huge productivity leaps, as long as the 'fun' of playing with the debugger doesn't get in the way

These don't look much like buttons, but they are. Selecting them by clicking will give a different dynamic menu.

When a breakpoint is set and the program has stopped, the display area contains a stack trace of the program at the point that it stopped. If you select variables from the source area by clicking on them, then their values will be displayed here. The display is updated as the contents change, so you can monitor values in variables as you step through the program. To set a value in a variable, you simply click with the middle button on the displayed number, get a vertical typing bar, type in the new number and off you go.

Working with the program

Underneath the display area is the *target menu.* This is a set of commands for controlling execution of the target program. The set of commands here are similar to those in Saber-C. You can start a program running until it hits a breakpoint. Then you can single step the program by source line using the `next` or `step` button. These buttons have much the same action except that the `next` button does not descend into routines. The double arrows at the end of the line allow you to adjust the relative areas being displayed by moving the central bar up and down the window.

Below the target menu are controls for the source area. The source area displays a particular file. In the example, the `#stop` line shows that a breakpoint has been set and triggered at that point in the program.

The `#stop` directive is added when a breakpoint is placed in the program, it has another interesting property - it adds an editable section into the program. You can click on the directive using the normal action that is used to start editing (middle mouse button). You can then insert any piece of C that you wish into the section. Ups contains a C interpreter that will find the inserted code and run it as if it were part of the program. This means that you can add selective tests into the source making debugging much faster.

Adding pieces of C is only intended as a debugging aid, so there is no question of writing the 'new' source back to the original source file. This can occasionally be a pain if you have developed some code to fix a problem by typing into the source window.

What don't I like?

It's certainly my debugger of choice. I wonder if it might be supplanted by Saber-C should that program be around on my machine for a little longer. Ups is simple enough to learn quickly, although there are some rough edges here and here. Some of the problems are due to the deficiencies in the approach. It is hard to get all the information from the source into the binary of the program. Sometimes you want to display the contents of structures for which you have no definition. This can be annoying.

Availability

The program supports much the same architectures as Saber-C does; the code runs on all Sun architectures and also the DEC VAX and DEC stations. The debugger also supports Fortran on the Sun 3, the Sun SparcStations and the VAX.

The program is in the public domain and if you have access to the Internet you can find it on various source archive machines. If you do not have access to this, you might call the author, Mark Russell at the University of Kent, Canterbury (0227 764000). Mark has said that he will arrange to distribute the code for a nominal charge as long as the demand is not vast. The compressed source is a little under a megabyte.

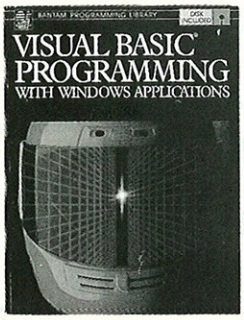
EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Books

The high and low roads to OOP.

Visual Basic - DIY style



Visual Basic Programming With Windows Applications is one of the first books on the eponymous Microsoft offering. (Isn't it funny how these product-based books seem to appear so soon after the release of the actual product. I suspect that there's a big market for books on pirated software...)

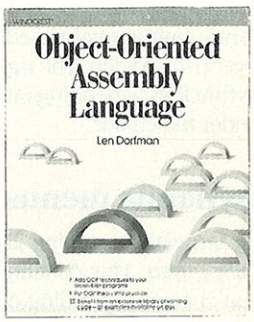
The first five chapters introduce the various nuts and bolts that make Visual Basic. These are used to build a complete application called 'Loan Calculator' which calculates the monthly payments on a bank loan and produces a table of results. After covering the installation of VB, *Visual Basic Programming* proceeds by introducing the reader to Loan Calculator, and adopts a step by step approach to creating this application. There is an entire chapter on placing controls on forms, and another on how to change the properties of these controls. The event-driven architecture of VB is explained quite well and a helpful guide to debugging is also given. Each phase in the development of the application is available on a 5 1/4" disk, bundled with the book, so that the reader can simply load each stage of the application without having t o type in all the source code.

The remainder of the book consists of eight example programs (also on disk). These are not covered to the same extent as Loan Calculator, although each one examines a particular VB technique eg *Sales Week* (I/O), *Address File* (Data structures and Control arrays), and *Program Summaries* (Dynamic Data Exchange). Each chapter contains a description of the application, screen shots of the forms used, a summary of the event handlers needed, a listing of the application and, finally, there's a description of how the given VB technique has been harnessed in the application code. Furthermore, *Visual Basic Programming* sprinkles little boxes ('Review Boxes') within the main text which it uses to introduce new VB statements. As the source code for the example programs is already on disk, I felt that the listings in the text were far too long (the 'Work Day' listing consumes 20 pages).

I cannot decide who would most benefit from this book. Although it adopts a tutorial style, the listings tend to use VB statements that haven't been previously explained (eg the use of the `switch` statement in the *Loan Calculator* listing). However it isn't a reference either (certainly no replacement for the genuine Microsoft manuals). Since this is one of the first books on VB, you may feel the urge to buy this one. I'd be tempted to wait.

Title: Visual Basic Programming With Windows Applications
Author: Douglas Hergert
Publisher: Bantam *Price: £42.99*
Pages: 472 *ISBN: 0-553-35317-9*

OOPasm addict



Object-Oriented Assembly Language by Len Dorfman gives an unusual perspective on the subject of object-oriented programming and design. The purpose of this text is to show how OOP techniques may be applied to assembly language programs, without compromising performance (the primary reason for choosing assembler). This is a book for do-ers more than readers in that the book is bulging with example code. It may also be viewed as a sequel to Dorfman's earlier book *Structured Assembly Language* in which the use of macros is shown to produce efficient but highly structured and readable assembly language modules.

In the introduction, Dorfman outlines the objectives of the book, followed by a brief debate on the benefits and disadvantages of OOP languages compared with assembly language.

Chapters one and two of the book are rather vague but provide a brief introduction to structured assembly language, demonstrating how objects may be constructed through the use of macros. The remaining chapters (three to eleven) are devoted to the documentation of practical examples. They describe how the techniques may be used to produce a basic set of objects needed to build simple assembly language programs that use screen and keyboard I/O facilities. The construction of several key classes (for text display, keyboard input, cursor handling and EGA graphic output) is described in detail and there is a great deal of example code written in 8086 assembler. However, the examples chosen are, to my mind, tqo simple, given the level of reader's knowledge assumed in the introductory chapters.

The objective stated at the beginning of the book was 'to add OOP techniques to your assembly programs'. In reality, the only object-oriented concept described (albeit the most fundamental one) is the basic class construct, with a brief mention of inheritance and polymorphism. This is all covered in the first chapter of just 10 pages. The book is a somewhat disappointing treatment of a topic which offers some exciting potential. For example, the design of classes appears totally intuitive, with no reference to data flow or object analysis techniques. Issues such as data encapsulation and preservation of object state are not mentioned, and there is no discussion of the proper construction of subclasses and polymorphic methods.

However, the fundamental principles proposed may be of real practical use to programmers, adding the benefits of object-based programming (readability, reliability and reusability), without the compromise of code efficiency common to dedicated OOP languages.

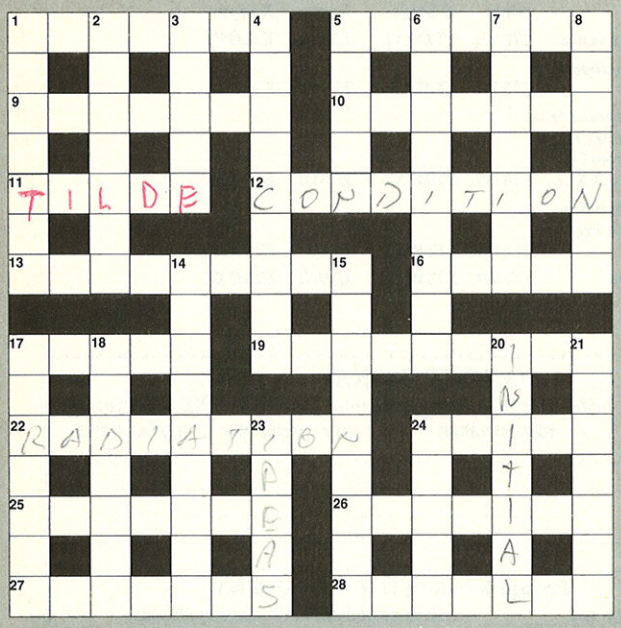
Title: Object-Oriented Assembly Language *Author: Len Dorfman*
Publisher: Windcrest *Price: £18.95*
Pages: 360 *ISBN: 0-8306-7620-1*

Review by Dr Martin Palmer

Books Received This Month

Prolog - From Formal Specification to Prototype by P Leadbetter, P Thomas and R Weedon	Blackwell Scientific Publications	£17.50	ISBN:0-632-03161-1	pp384
Programming in Ada plus Language Reference Manual by J G P Barnes	Addison Wesley	£27.95	ISBN:0-201-56539-0	pp813
Newnes C Pocket Book by Conor Sexton	Butterworth-Heinemann	£12.95	ISBN:0-7506-0221-X	pp304

OCTOBER .EXEWORD



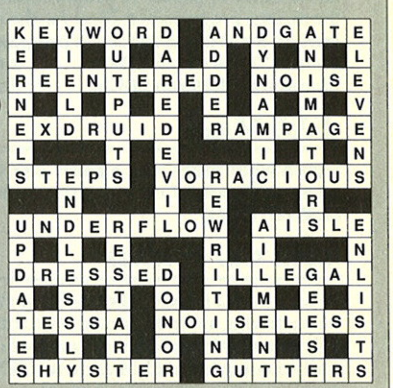
ACROSS

1 It converts input to output (7)
5 Flow of charge for the time being (7)
9 Act on data item in 8? (7)
10 Groves of training places? (7)
11 ASCII 126 (5)
12 Linguistic object like IF/ENDIF (9)
13 Share of programs' sales (9)
16 Sibling's child up the tree (5)
17 Description on package's packet (5)
19 Row of tips (9)
22 Hazard from the screen or source of channel (9)
24 Output size measure (5)
25 Final source of ink jet (7)
26 Ask of a database (7)
27 Babbage's systems (7)
28 I11? Rest round by the grid (7)

DOWN

1 Output unit in conspiracy? (7)
2 Program section going on a second (7)
3 Get rid of hesitation as in the orient (5)
4 Making a plan on a 1ac? (9)
5 Data structure sequence on the printer (5)
6 It cuts down ac (9)
7 This magazine, being attractive, may run (7)
8 Uneven rate in the site of operation (7)
14 One who deals with data issues and returns (9)
15 A computer one is a technologist really (9)
17 EAN way to behave in a pub (7)
18 Nemesis' way out of a routine? (7)
20 Value at the start of the run (7)
21 Respects and meets up in two ways (7)
23 Intangible assets of analyst and programmer (5)
24 Irritation at the 24ac it seems (5)

'.EXEWORD' compiled by Eric Deeson



SEPTEMBER .EXEWORD

.EXE directory

This is a selection of products from the .EXE directory. Call NOW for a complete catalogue, or to place your order direct.

0442 824501

.EXE Disks

A collection of specially selected source listings from .EXE, and a selection of Public Domain and Shareware utilities. Each disk contains around 1.4 Mb of data and an average of 15 files per disk. Some of the contents are given below.

Volume 4 - ref: ED45 - 5.25" or ED43 for 3.5" disks
BOSS A superb text windowing package for C. It lets you create, write to, move and generally manipulate text windows. It also lets you create pull down and pop-up menus.

Volume 5 - ref: ED55 - 5.25" or ED53 for 3.5" disks
CTASK20.LZH CTASK V2.0 is a multi tasking kernel for C. It's a complete task schedule for MS-DOS that allows you to run parts of your C program in parallel. MS-DOS call can also be made concurrently.

SNAP402.LZH Version 4.02 of SNAP, an excellent dBASE code documentor which now handles dBASE IV and FoxPro too.

Volume 6 - ref: ED65 - 5.25" or ED63 for 3.5" disks
INTER290.LZH Source code, in C and assembler, to LHARC, the file compression/decompression program as used on many of the .EXE disks.

SCANV64.LZH A virus scanner which will scan your hard disk and search for around 50 known viruses.

Volume 7 - ref: ED75 - 5.25" or ED73 for 3.5" disks
CDRIVER.LZH A collection of files that explain how to write MS-DOS device drivers in C.

4DOS301.LZH 4DOS is a complete replacement for COMMAND.COM. 4DOS ads a number of new features, including command aliases, command recall and an extended DOS batch language.

Volume 8 - ref: ED85 - 5.25" or ED83 for 3.5" disks
LHARC- compression software, for packing & unpacking programs. **BACKMENU-** throws up a pop-up menu with selection of launchable programs whenever desktop is clicked. **CV-** CodeView for Windows. Complete source & assembly level Windows debugger. **DDEWATCH-** debugs DDE programs. Lists, translates, & logs all DDE calls while in background. **WINPOST-** Post-it notes for Windows. **WINSTUB-** better DOS stub program. Doesn't print error message when Windows programs run from DOS, automatically executes Windows, & invisibly re-runs application.

Volume 9 - ref: ED95 - 5.25" or ED93 for 3.5" disks
LHARC- See Volume 8. **BACKISS-** Back issue list for .EXE. From Vol 1, No 1 to Vol 5 No 9. **DPMI-** DOS Protected Mode Interface **GNURCS-** GNURCS is a DOS port of GNU's Unix RCS. **SID-** DES encryption utilities. Source & executables for entire DES implementation. **SVGA-** library for 256 colour Super VGA adaptors.

Volume 10 - ref: ED105 - 5.25" or ED103 for 3.5" disks
LHA- version 2.1 - public domain compression program (was LHARC). One of best speed & compression ratios of any compaction program. **LHSRC-** full source to LHA program. **BOOKLET-** shareware program, similiar to PR2, allows 80-column manuals reprinted as A5 booklet.

BOYMOORE- fast search of strings. Full implementation of Boyer-Moore algorithm. **DOSTRACE-** a TSR utility for 386 machines monitors & lists interrupt calls, open files, watches addresses, dumps memory, breaks on software interrupts. **INFOPLUS-** relays information on your PC. Turbo Pascal source shows how it works. **INTER191-** Famous Interrupt List. **MOUSEQ-** queues mouse interrupts. Full C source. **POPBUG-** newly discovered 386 bug, affects POPAD instruction. Test code & description. **PR2-** ANSI C program, prints 80-column text files in 2 columns on certain Epson compatibles. **SPAWNO30-** Replacement spawn function for MS & Turbo C swaps a running program to disk, EMS, or XMS memory while executing a child program or subshell. **STATEMAC-** Ray Jones' state machine interpreter in C.

.EXE Library

SPECIAL OFFER

Even the best programmer needs a good set of reference books by his side. We have bundled a few books with collections of bound articles from .EXE. substantial savings can be made by ordering the two. All books are available separately. Please call 0442 824501 for individual prices.

Intensive C Course Book - illustrates the principles of good programming style and sound design. Available @ **£10** with the collection of articles on C from .EXE. (Would normally be £15.20). ref: **EBAC**

Liskins dBASE IV Programming - gives plenty of applications to demonstrate the elements of dBASE from fundamentals to applications design. Available @ **£28.95** including collection of articles on dBASE (normally £36.45). ref: **EBAD**

The Master C Book & Disks - A comprehensive, High Powered book and software package which revolutionises the instruction of C. 4 disks are provided which automatically guide you through C topics. An excellent way to learn. **£39.95** ref: **EBMC.**

Professional Programmers Guide to Pascal - tutorial and reference material are carefully integrated. Standard and Extended Pascal are both covered in detail. Available @ **£11.00** including a collection of articles from .EXE on Pascal programming. (Normally £14.49). ref: **EBAP.**

The Professional Programmers Guides adopt a no-nonsense approach to programming and are designed for the serious programmer. Ideal for the novice or as an aide memoir for a\the professional programmer.

Fortran 77 - £8.99 ref: EBKF **Ada - £9.99 ref: EBKA**
Modula-2 - £8.50 ref: EBKM **Pascal - £8.99 ref: EBKP**

.EXE Mugs

Stylish white mugs with a green .EXE logo - guaranteed to be of use every day in the office!
£4.50 ref: **EM1W**

.EXE Car Sticker

White oval car sticker (like the GB stickers used when driving abroad) printed with the .EXE logo - only other .EXE readers will know where you're really coming from! **£1.50** each. ref: **ECS1**

.EXE T-Shirts

.EXE T-Shirts, featuring the .EXE Logo or one of 4 other slogans will make you the envy of the office. Available in either white or dark green in Large **£9.00** or Extra Large **£9.50**

Reference number - please quote when ordering:

	White		*Green*	
LOGO	*Large*	*X Large*	*Large*	*X Large*
Born to compile	*ET1WL*	*ET1WXL*	*ET1GL*	*ET1GXL*
.EXE Developers do it best	*ET2WL*	*ET2WXL*	*ET2GL*	*ET2GXL*
'As he returned from the computer room' (cartoon from Oct '89 issue of .EXE)	*ET3WL*	*ET3WXL*	*ET3GL*	*ET3GXL*
'Verity Stob has NOT got a big bottom'	*ET4WL*	*ET4WXL*	*ET4GL*	*ET4GXL*
.EXE Logo	*ET5WL*	*ET5WXL*	*ET5GL*	*ET5GXL*

ORDER FORM

I would like to order the following products from the .EXE directory:

DESCRIPTION REF. NUMBER QTY UNIT PRICE TOTAL PRICE

Postage & Packing £1.00 TOTALPRICE

My total order value is £_____ *(All price include VAT where applicable)*

I would like to pay as indicated (please tick): ❑ Cheque enclosed

❑ Please debit my VISA/Access card No:

Card No: _____ Expiry: _____

Name: _____

Company: _____

Address: _____

_____ Postcode: _____ Tel: _____

Signed: _____ Date: _____

Please return to Process Communications Ltd, PO Box 388, Tring, Herts HP23 4EB.
Please quote reference numbers when ordering goods.

G.I.S. – The Technology of the 1990's

The U.S. National Science Foundation defines a Geographical Information System as "A computerised database management system for the capture, storage, retrieval, analysis and display of spatial, locationally-referenced data."

What they don't say is that G.I.S. incorporates almost all of the emerging enabling technologies including "Open Systems standards", OOP and OOD techniques and Hypermedia as well as more conventional RDBMS and 4GLs. G.I.S. is growing and your career could grow with it!!

Oracle/Ingres Database Designer

£24,000 **Hertfordshire**

This solutions vendor markets Arc/Info, the world's most successful G.I.S. tool-set. You will be developing new database software to enable their large customers to integrate multi-vendor DBMS into sophisticated corporate information systems. Experience of CASE tools is essential.

Software Developer

£20,000 **C.London**

A new appointment with a market leader in geo- demographics and market analysis applications. Your mission will be to develop new software modules for the UNIX market. Fluency in 'C' is essential, knowledge of 'C++', X-Standards and SunOS is an advantage.

G.I.S. Customer Support

£17,000 plus car **Surrey & Cambridge**

Two new positions with leading G.I.S. vendors, both seeking good inter-personal skills and a year or more of experience in Utilities or Local Government applications. Technical skills should include UNIX, 'C' programming and a Macro command language.

Young UNIX/G.I.S. Programmer

£20,000 **Holland and Germany**

This U.S. Systems Integrator dominates the digital mapping industry. Your role will be to work alongside European customers in the utilities industry in order to develop new applications in 'C' and FORTRAN 77. Graphics and database experience is very desirable.

Software Engineers

£21,000 **Cambridge**

This 3 year old company is the G.I.S. industry's innovator. They have technology transfer agreements with academia enabling them to propagate research into 5GLs, Object-oriented techniques and version-managed database technology. Your skills must include 'C', 'C++', UNIX and graphics.

Programmer/Analyst

£16,000 to £18,000 **Thames Valley**

If you hold a good Computer Science degree qualification and you have worked in industry for at least a year then your skills could be cross-trained to this application of G.I.S. They use Sun Sparc workstations under SunOS and 'C' for development.

To apply for one of these positions, or to learn more about G.I.S. contact ALAN CARNELL at Concurrent Appointments, the G.I.S. Recruitment Specialists.

Concurrent Appointments
Software Recruitment

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

SOFTWARE PROFESSIONALS

SURREY **to £18k**

Leading transport consultancy seeking science, computing or mathematics graduates with **VAX/Fortran** or **'C'/UNIX** experience for new projects in autoguidance and route mapping.

HANTS **£15k-£22k**

Intelligent network developer seeking bright engineers with an interest in datacomms and at least 2 years' experience of **VAX/VMS, 'C'** and **68000** assembler. Any of the following useful: **Windows, X.25, X.400, OSI, SNA X.75** or **C++**.

YORKS **to £20k**

Widely acknowledged as leaders in the **Document Image Processing** market, our client is seeking experienced **'C'** and **MS** or **X-Windows** software engineers. Any **Oracle/Ingres/SQL, UNIX, TCP/IP** or **Streams** useful. Bonus, BUPA, Pension and relocation offered.

MIDDX **£18k-£25k**

U.S. geoscience developer is seeking a scientific software programmer with proficiency in **'C'/UNIX**, to specify, design and implement a seismic data interface package. Any knowledge of **Graphics, X-Windows** or **VAX/Fortran** is useful.

S.HANTS/I.O.W **£neg.**

Varied and interesting work in the areas of **Process Control, CIM** and **Information Systems** if you have good **UNIX** and **'C'** experience, Waterfront location.

BELGIUM or ITALY **to £25k**

International software house seeking **'C'** programmers with experience of two or more of the following: **MS-DOS, MS-Windows, Pascal, Oracle, UNIX or X-Windows.**

BUCKS **£25k-£35k**

Revolutionary emulation software product developer is seeking a **Software Project Manager** to direct and manage a number of specific projects, if you have good PC experience encompassing some of the following: **UNIX, DOS, Macintosh** or **X-Windows.**

SURREY **£15k-£22k**

Analyst/programmers and programmers needed by leading accounting organisation if you have three of the following skills: **UNIX, 'C', Informix, Prolog, Oracle or ProC.**

W.COUNTRY **£18k - £25k**

Systems Consultancy in the field of public utilities, government, legal and the pharmaceutical industries is keen to recruit good calibre graduates with some of the following skills: **'C', UNIX, Paradox, RDBMS, CASE, SSADM, Prompt, Prince, Yourdon, MS-Windows, MS-DOS, OS/2, Presentation Manager, X.25, SNA, VAX** or **Networking.**

BERKS **£neg**

The worlds leader in communications development is sseking Software Engineers with **VAX/VMS, Pascal** or **'C'** experience. Any of the following would be useful: **UNIX, Fortran, CCITT** or **X-Windows.**

HERTS **to £20k**

Leading force in the development of Printing Control Systems seek a real-time software engineer with most of the following: **'C', Assembler, M68000, Z80, MS-DOS, OS/2** or **Hardware Interfacing.**

BEDS **£17k-£22k**

Exciting seismic exploration project requires real-time software engineers with most of the following: **'C', UNIX, X-Windows, Motif** or **TCP/IP.**

HERTS **to £23k + Car**

3D graphic tool developer running on Apollo, Silicon Graphics and IBM RISC/6000 workstations is seeking **'C'/UNIX** programmers. Any experience of **Fortran, C++** or **CAD** useful. Excellent promotion prospects.

SURREY **to £20k**

Premier publishing software developer seeking talented software engineers with experience of **'C'/UNIX** or work on graphics projects utilising **SUN** and **Macintosh** workstations.

E.ANGLIA **£ Excellent**

Internationally renowned comms developer seeking graduates/post-graduates for exciting projects in Network Management, Videotex, Office Automation, Multi-media and PABX. You should have at least two of: **VAX/VMS, 'C', UNIX, C++, MS-DOS, IKBS, M68000, Pascal, Oracle, DPNSS** or **SUN.**

WILTS **£15k-£24k**

Software Engineer and **Development Manager** sought for small teams developing **'C'** on **UNIX** under **MS-Windows.** Any other experience of **C++, X-Windows** etc useful.

For further information about these and MANY other opportunities either telephone

081 876-0102 / 081 392-1514

or write to; ACUMEN SEARCH & SELECTION INTERNATIONAL
EBC House, 1a Ranelagh Gardens,
London SW6 3PA Fax: 071-371 6502

ACUMEN
SEARCH & SELECTION INTERNATIONAL

ASH ASSOCIATES

We specialise in the Recruitment of Software Design and Support Engineers in the South East for Real Time Applications including GRAPHICS, COMMUNICATIONS, CONTROL/ROBOTICS, SIGNAL PROCESSING & MODELLING

We URGENTLY seek highly qualified design engineers for a BERKSHIRE based client, Salaries £20K PLUS.

UNIQUE OPPORTUNITY

Lively, outgoing Software Design enthusiasts with C++ design expertise are urgently sought by a Major International company to work on their unique applications for AI and Expert Systems.

They seek in addition to your C++ experience knowledge of UNIX, OSF/Motif and X-Windows gained within a design environment.

YOU will enhance your career and future prospects working within this High Technology Application area with the backing and benefits of this expanding and World Leading Company.

Call James Hunt or Ron Cook NOW!
TEL: (0425) 475480 (24hrs)

ASHassociates

Recruitment Consultants
3 Pipers Ash, Ringwood, Hants, BH24 1UF
Tel: (0425) 475480 Fax: (0425) 480807

PEOPLE IN UNIX*

*Unix is a trademark of AT&T

ACTIS
is seeking people with good technical experience of **OPEN SYSTEMS** which is still a growing sector of the IT market.

We have a number of instructions in the North and Midlands and we are looking for people with specific skills in:

Networking and Communications,
Relational Database/SQL
4 GLs, CASE, IPSE
C and WINDOWS
Accounts/Commercial and Scientific Programming
Training and Documentation
Technical Sales

We are seeking consultant, project management, development and support staff from graduate trainees to those with ten years experience.

To discuss these and other opportunities talk to **Honor Lindsey** on **(0204) 20200** or send your CV to **ACTIS RECRUITMENT**
17 CHORLEY NEW RD, BOLTON BL1 4QR

ACTIS
PEOPLE IN UNIX*

C/C++/Windows/Unix

Cambs **to £30K package**

This young, dynamic, professional international consultancy are expanding because of an increasing demand for their services. You will be working with a team of IT professionals developing and implementing a major MIS system within a C++, WINDOWS, SYBASE, UNIX environment. The successful candidates will be educated to degree level with at least 2 years experience in a C/C++ and WINDOWS environment. Additionally, they are seeking a UNIX consultant for providing tuning, porting and kernel level expertise.
Contact: Christine Trybus Ref: CT4001

'C' MSDOS

Yorkshire **to £20K**

Do you enjoy a challenge and can you respond to a stimulating and varied environment? Here is a chance to make real use of your existing skills as well as acquiring new ones. Ideally you will have a minimum 18 months C in a MSDOS/ UNIX environment with a knowledge of C++, Basic, 4GLs, SQL (preferably Paradox). This is a superb opportunity to enhance and develop your software skills on new and exciting projects.
Contact: Paul Innes
Ref: PI4001

C and OS - 9/68000

Cotswold/Wilts **to £20K**

Join these small, friendly teams and design bespoke software for graphics/communications. This young company offers you early technical responsibility in real-time applications and the chance to take account of the latest advances in navigation, communications and display systems. Be there at the start of this new project and be sure of your future with their full order book. For further details on their exciting product range and how these opportunities can benefit you,
Contact: Teresa Maddern Ref: TM4004

1st Class Hons

London **to £22K**

This top class consultancy are seeking 1st Class Hons graduates with at least 1 years experience in the following areas:

*** 'C' * Oracle * Ingres**
*** Case Tools**
*** Structured Methods**

You will be working on a variety of challenging projects using the latest software tools. The company offers superb training and the opportunities to develop your career are limitless.
Contact: Christine Trybus
Ref: CT4002

For more details

Contact the appropriate consultant for details on the above and many other vacancies on **0442 231691** days or 0442 69740 eves/wkends. Alternatively write to:
Executive Recruitment Services, Hempstead House, Selden Hill, Hemel Hempstead. HP2 4LT
or fax CV to 0442 230063.

* STOP PRESS *

to £30K **Middlesex**

Urgent requirement for ORACLE, Analysts and Database Administrators to work with version 6.
Contact: Christine Trybus Ref: CT4003

DEVELOPMENT SPECIALISTS

SQL/Windows Development

The City to £33,000 + benefits

A major City name is currently looking to expand its dealing room development team with one, possibly two, support analysts. Working closely with the dealing staff, the successful individuals will be required to formulate and document requirements and then design, develop and implement applications. A numerate graduate, you should have at least 3 years experience within a similar environment and have development experience using an *SQL* (preferably *SQLBase* and *SQLWindows*) in a *Windows* environment running a *LAN*, preferably *LAN Manager*. This highly pressurised role will require a flexible, decisive attitude to complement excellent technical skills. In return you will be offered the opportunity to work in a rewarding environment and will receive a generous package with excellent benefits.

Ref: PCEX12/3

Software Engineer

Central London £17–24,000

This young, dynamic company specialise in advanced software solutions for the international dealing room environment. Based in Central London they are now expanding their applications area and are looking to recruit 3 software engineers. As a member of this team you will be responsible for the design, implementation, testing and installation of new and existing software. The ideal candidate will be educated to degree level and have 2+ years commercial experience with exposure to clients/users. Essential technical skills include 'C' *development* under *DOS* or *OS/2* and preferably experience of the *SQL* language, Multitasking, Transaction Processing or Workstation User Interfaces. Knowledge of FX, Money Markets or Off Balance Sheet environments would be advantageous. The work is demanding both technically and in application knowledge, the working atmosphere is both dynamic, informal and friendly. *Ref: PCEX12/4*

DOS and 'C'

London to £18,000

An established developer of communication systems, our client illustrates its continued success by currently expanding its development team in a difficult economic climate. If you are a 'C' programmer, with at least 18 months development experience under *DOS*, you could take up this exciting opportunity and join a dynamic team, working on fast moving projects in the communications environment. The ideal candidate will be a team player and yet self-motivated, have initiative and first class interpersonal skills. Fluent in 'C' *programming*, any exposure to *Pascal*, *Novell* and communication systems development, particularly message switching would be extremely advantageous.

Ref: PCEX12/2

Unix & Informix Developer

Middlesex to £20,000 + car

An *Informix* and *Unix* analyst/programmer is required to join a young and enthusiastic development team. Full project management is on offer, involving end-user analysis, programming, implementation, testing, support and training, as well as the supervision of programmers depending on the size of the project in hand. A graduate in computer science would be preferred – but is not essential – and you will be in your mid to late 20's. Projects will be varied providing challenge, and a demanding environment. In return, you will commit your adaptability and professionalism to the job and will be rewarded by a competitive remuneration package. Excellent prospects.

Ref: PCEX12/1

For further details of these and other permanent vacancies, please contact Conrad Hills, quoting the relevant reference, on 071-734 4010 (office hours) or 081-542 8724 (evenings/weekends). Alternatively, write to McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB or fax your CV on 071-734 1297.

mcgregor boyall

IT HUMAN RESOURCING

FOR PERMANENT AND CONTRACT IT RECRUITMENT TELEPHONE 071-734 4010

WEST YORKSHIRE

COBOL Analyst Programmers with UNIX experience, future ORACLE training. *to £15,000*

Software Engineer, minimum 1 year 'C' with 68000 and 8086 Assemblers. *£13,000*

Electronic Design Engineers *£20,000+*

UNIX Systems Admin. *£12,000*

Analyst Programmers (min. 8 years experience) 'C' and UNIX *£18,000*

If you are a Programmer, Analyst Programmer, Software Engineer, Electronic Design Engineer or UNIX Systems Admin. seeking the environment best suited to your next career move in West Yorkshire then telephone Vincent Atherton on Leeds (0532) 504560 or write to:

AIREDALE RECRUITMENT

Realtex House, Micklefield Lane,
Rawdon, Leeds, LS19 6AX

AIREDALE RECRUITMENT

PROGRAMMER (WINDOWS 3.0)

Buckinghamshire
Negotiable Salary

This very successful manufacturer of computer-based scientific instruments is part of a high-profile British group and a world-leader in its field. They require a Programmer who will liase closely with application specialists in the implementation of state-of-the art computer-based analytical products. To be relevant, candidates must have at least two years experience of writing Microsoft Windows applications in 'C'. Ideal candidates will also have had a scientific background and an understanding of real-time controls together with assembly language programming skills. Self-reliance and an ability to work to tight deadlines are also important.

HRS

Contact Lionel Eras at
Human Resource Services (UK),
58 Windmill Road, Hampton Hill,
Middx., TW12 1QU
Tel: 081 941-2686 (24 hrs)
Fax: 081 941-1176

Chess Computer Services Ltd

Search, Selection and Advertising Consultants

Park House, Greenhill Crescent, Watford Business Park, Hertfordshire, WD1 8QU Tel: 0923 225363 Fax: 0923 225051

DEVELOPMENT

SENIOR UNIX SOFTWARE ENGINEERS - £30K + Car + Bens.
With an intimate knowledge of the UNIX operating system down to kernel level. This industry is recruiting UNIX specialists with strong C programming skills. Numerous LAN support vacancies for recruits with experience in one of the following - to £35K: LAN Manager Novell Lan Server Vines, TCP/IP, UNIX

INFORMIX SENIOR SYSTEMS DESIGNER - To £30K
Top city based consultancy need good Informix, C and UNIX skills to oversee a variety of large projects.

WORLD CLASS NETWORKING SOFTWARE ENGINEERS C£25K
To design and develop LAN products, working in C & Assembler.

SYBASE - ANALYST PROGRAMMER £18 - £25K
Top Comms company require analyst programmer to develop in-house database.

PARADOX - ANALYST PROGRAMMER - £20K
With in-depth knowledge of PAL. Any financial or retail experience desirable.

AS400 SENIOR ANALYST PROGRAMMER - £25K
RPG 400 analyst programmer to lead a team of four within this major vehicle manufacturer.

C LAN PROGRAMMER - £25K
Experience of working in a DOS environment, OSI/X.25 skills

C++ PROGRAMMER - £25K
Ambitious graduate calibre developers to join this international software services company.

UNIX SOFTWARE ENGINEER - £25K
Working to Kernel level porting applications to different hardware platforms.

UNIX SUPPORT - £20K + Car
UNIX and Uniplex experience to configure, administer and customise Uniplex for customer requirements.

CLIPPER ANALYST & PROGRAMMERS - To £28K + banking benefits
Merchant bank requires junior and senior Clipper professionals for back office database development.

C PROGRAMMERS INTERNATIONAL BANK - £25K + bank benefits
To develop front end dealing room systems for this prestigious institution.

TOP CLASS SOFTWARE ENGINEERS to £25K
To design and implement network drives working in C and Assembler on local and wide area networks.

SALES EXECUTIVES - £25 - 30K Basic 60K OTE + Car
Major software house needs professionals to sell into corporates and financial institutions. Must have knowledge of IBM mainframe environment and be able to negotiate at top levels.

SALES EXECUTIVES - £22 - 27K Basic £50K OTE + Car
To sell UNIX applications to VARs and major accounts. A knowledge of data comms useful. Positions available UK wide.

IF YOU HAVE EXPERTISE IN THE ABOVE OPPORTUNITIES OR YOU ARE SIMPLY A COMPUTER PROFESSIONAL LOOKING FOR A NEW POSITION CALL NOW FOR VACANCY DETAILS IN SALES SUPPORT DEVELOPMENT

RONICOM
RECRUITMENT

5-7 Sedley Place (off Oxford Street), London W1R 1HH
Tel: 071 491 3640 Fax: 071 499 2546

OS2 DEVELOPERS **up to 25k** **London**
This quality Software house is looking for good quality Software developers who have PM, C, Windows SDK skills behind an OS2 background: very rewarding!

ANALYST / PROGRAMMER
up to 35k plus bens **London**
Qualified to degree level 2 yrs plus experience, SQL-Windows, good opportunity to join an elite financial Software Consultancy, new development lots of scope.

SYBASE **up to 30k** **London & Home Counties**
Six months or more experience, gives an entry to an increasing number of blue chip clients offering a variety of challenging opportunities at all levels.

MAC DEVELOPERS **up to 20k** **London**
NOT DTP! We are representing a client who needs system level MAC skills and some application knowledge of the likes of OMNIS or 4th Dimension is useful.

ORACLE SQL*FORMS 3 **up to 30k plus car** **London**
A growth Software Consultancy requires strong Analysis and Design skills, CASE an advantage. Team leading emphasis.

BIOS SOFTWARE ENGINEER **up to 18k** **Surrey**
Low level, Assembler, TASM, MASM, device drivers and a good mixture of Hardware and Software for this esteemed Manufacturer.

SOFTWARE DEVELOPER **up to 20k** **Cambs**
C, UNIX and X-Windows programmers required for a recognised Software House. 1-2 years experience, additional expertise on PCs or Workstations is an advantage.

INGRES/ORACLE/SYBASE **up to 25k** **London**
You will have had experience in large projects and SSADM within an established consultancy. Preferably Systems House experience.

ORACLE/TECHNICAL SERVICES **up to 25k** **London**
A Technical Support/Consultancy role, systems development, installations, on-site advising, in-house administration. Platforms are VMS/PCs/Networks.

WINDOWS/C/SECURITIES **up to 22k** **London**
Either PC or UNIX platform knowledge, database experience would be an advantage for this C/Windows development role in a Securities environment.

C SOFTWARE DEVELOPER **up to 18k** **London**
C - Assembler - LAN - Real time and VGA card skills. Good hardware expertise and structured techniques would be a bonus.

C LAN PROGRAMMER **up to 18k + Bonus** **SW London**
Competent in DOS environment, OSI/ICL networking skills, X25.

DOS/LAN SENIOR ANALYST PROGRAMMER
21k + Bens **Surrey**
Fully competent in C - 3-4 years experience, network programming, data communications, and a few successful software implementations under your belt.

SOFTWARE PRODUCT SUPPORT SPECIALIST
18k + Car + Bens **Berks**
International support, providing solutions to queries, Assembler, Cobol, Workbench and MSDOS and MVS.

CHIP SET DESIGN CONTRACT **Good** **Sweden**
ASICs, low level voltage, experience of vendor toolkits - Synopsis, Cadence, Varilog, Edge.

OS2 TECHNICAL SUPPORT **18k** **W London**
Must be experienced in Software House environment type products, Presentation Manager or Windows plus an understanding of 4GL applications.

4GL PROGRAMMER **16k** **N London**
IS2 and C languages required for Development and Support of Financial Systems.

INFORMIX PROGRAMMER **16k** **Berks**
2 years experience of this 4GL for work on development of, among other things, a new membership package.

PRODUCT EVALUATION **18k Neg upwards** **Berks**
Technical Support background ideal for this position to evaluate and assess performance and compatibility of PC and related components.

SUN SYSTEMS EXPERIENCE **15-25k** **W London**
2 positions requiring this platform. Systems Programmer for large end user, which includes some Systems Admin. Also Field Service/Tech Support role for Sun/SunOS environments.

INGRES ANALYST/PROGRAMMER **up to 20k** **Surrey**
2-3 years experience for work on large international projects

SOFTWARE DEVELOPER **up to 25k** **S London**
C, PCs, UNIX. Previous experience in EIS systems a distinct advantage.

UNIX SYSTEMS ADMINISTRATOR **18k** **London**
Previous experience of running in-house system for large end user would be nice.

LOW LEVEL DEVELOPMENT **20k** **N London**
68000 Assembler, Chip Sets and Electronics development for large Manufacturer. Knowledge of Communications and Digital techniques is also desirable.

RM-COBOL **Neg** **London**
Skilled Software Developer for a Support Consultant role, must have RM-Tools & some OOP's or C++ could be useful

SMALLTALK DEVELOPER **Neg** **NW London**
Artificial Intelligence environment, good OOPS and C with Smalltalk. Leading edge Software House.

EIFFEL SKILLS **Neg** **London**
Any development environment, preferably from a Systems & Software House background. Good experience and skills base essential.

FOXBASE/FOXPRO **16-20k** **Oxon**
Technical Consultancy and Development role for a renowned Software House. Suit second jobber but must have solid development experience.

C/WINDOWS/IMAGE PROCESSING **up to 25k** **London**
A leading light in the independent comms arena is looking for a skilled software engineer to set up a new project.

CONTRACT-CONTRACT-CONTRACT-CONTRACT-CONTRACT!

WINDOWS. MSDOS. C.	**6 months**	**City**
DEC FORMS. RDB. VAX BASIC.	**4 months**	**City**
ORACLE. SQL.	**3 months**	**London**

These are only a small selection of what are currently available. Please do not hesitate to give either Mike Dearing or Simon Gudgeon a call on 071 491 3640, or after hours on 081 767 1003 or 0621 773106.

ADVERTISERS INDEX

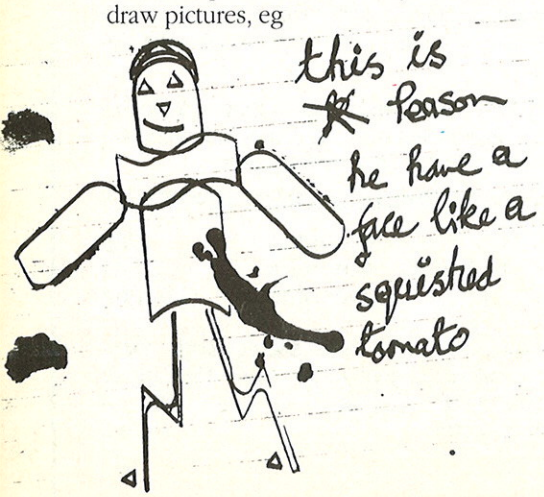
STOB - Wot any Bule kno

While listening to his original exploits being Griff Rhys-Jonesd on the radio recently, Stob wondered how nigel molesworth would cope with the modern computerised classroom.

Comp is just like Fr, Lat, Geog geom ect ect eg it is totaly wet and weedy. It is tort by maths masters, who say now boys, we are going to do something v exciting, compleetly different from alg, and then rite

LET A + B = 2 * C

on blakboard, chiz, chiz. The only thing to do in comp lessons is to use your stensil to draw pictures, eg



Most comp is about Bule, who was so cleva he invented bule alg (sick) *200 yrs befor ther was any use for it.*

Scene: brekfast table of Mr & Mrs Bule, 1791

Mr Bule: IF it is raining, THEN I shall NOT take the dog for a walk.

Mrs Bule: Pass the marmalaid please dere.

Mr Bule: Oringe XOR lemon-and-lime flavor?

Mrs Bule: I do not mind, sweethart.

Mr Bule: This IMPLYS you like oringe AND you like lemon-and-lime.

(Enter 2 men in wite coats with strait jaket)

Mrs Bule: Hes over there.

(The men grab Mr Bule and carry him away kicking and screaming)

Mrs Bule: Piece at last.

Sometimes we have a comp practikal. The skool comp is a BCCI B, three trillion years old and hav about half the power of Grabber's calc and game watch. (Grabber is head of skool, v rich and winner of the Mrs Joyful prize for algorhythms). The key board is full of chewing gum, bungy ect, so when you try and type it go

FORRRRRRRRRR I=11111111 to 3

Not even the master can work it, every time he program it, it sa

SYNTACKS ERROR LINE 3
RELODE DISK AND TRY AGAIN
YOU ARE A TOTAL FAILURE

and we all yell Yah Booh Sucks! sir couldn't program a fiver out of a cashpoint. One time molesworth 2 got this grate game called gulf war 3 which was smashing NEEEEEOW ZOOOM dagadagadaga BOOM but the beak confiskated the disc, chiz.

The only good thing about comp is that it enable you to IMPRESS GROWN-UPS. Mater sa: nigel, could you set the video to record the new mini-series with david mcullem and that girl from dallas, yes, he is cleva isnt he, your so lucky to do comp at skool, nigel, and could you program the micro wave to heat up the leftovers please darling?

If it goes on like this, by the year 2000 no grown-up will be able to open a door. And what will they all do then, pore things?

.EXE

Opportunities for Software Professionals

SYSTEMS PROGRAMMER

BERKSHIRE **To £20K**

1-2 years COBOL experience within software tool development or applications programming. Knowledge of PC DOS and OS/2 with presentation manager. To provide product development and support skills for important COBOL tools.

Ref: 01 - 9/91

SOFTWARE PROGRAMMERS & ANALYST AS400/ IBM - SYS 36/38

MOST AREAS **To £25K**

We have clients seeking experienced Software Engineers who have worked in an AS400/IBM mainframe environment. Positions range from programmers through to Senior Analysts and Support personnel.

Ref: 02 - 9/91

SYSTEMS ENGINEER

HAMPSHIRE **£Neg**

To provide a problem solving function in development department and generate high integrity solutions to difficult problems. Minimum experience to include five years hardware/software problem solving, six months low level programming, knowledge of classical data structures, 1st Honours Degree, 22-30 years old

Ref: 04 - 9/91

SENIOR SYSTEMS ENGINEER

HERTFORDSHIRE **£18-22K**

Our client is engaged in a continuing programme of development and support of microprocessor based real-time data acquisition and analysis systems of PC-based support packages. They are seeking an experienced Systems Engineer - in both hardware and software - and their combination into successfully integrated systems - to manage and take part in the engineering activities of several new projects. Must have an electronic engineering/scientific background and proven skills in project management.

Ref: 05 - 9/91

TECHNICAL SUPPORT

THAMES VALLEY **£Neg**

Various clients are seeking young, enthusiastic, goal orientated personnel, willing to work in a fast moving environment. Must have two to three years experience in either LANS, Windows, UNIX, EXCEL with a Degree or equivalent.

Ref: 06 - 9/91

SENIOR PROGRAMMER

LONDON **£17-23K**

1-3 years experience of at least 2 of either IBM P52/752, C programming language/windows, and experience in presentation manager Easel.

Ref: 03 - 9/91

68000 PROGRAMMER

HAMPSHIRE **£Neg**

Responsible to take high level descriptions of functional requirements, undertake the design and produce the design documentation, implement and test the solution. Must have 68000 Assembler programming experience (with some 'C') and have worked in a networked PCAT development environment. Minimum requirements include six months low level programming experience, low level programming of True Real Time tasks, 2:1 or 1st Honours Degree, 22-30 years old. *Ref: 07 - 9/91*

SYSTEMS PROGRAMMER

BERKSHIRE **£13-15K**

To provide programming and maintenance support for end-user products in the IBM PC market. Experience of 8086/80286/ 80386, Assembler, DOS or OS/2 for creation of COBOL code generator. *Ref: 08 - 9/91*

PROGRAMMER

LONDON **£16-20K**

Must have minimum of one years experience of Oracle including six months of Oracle* Forms and/or C programming language. Degree level qualification, PC programming experience and city-based applications. Responsible for writing programs, creating test schedules installation and diagnosing faults on site. *Ref: 09 - 9/91*

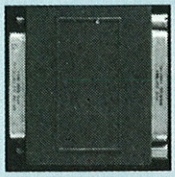
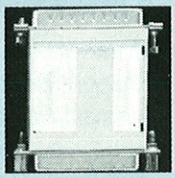
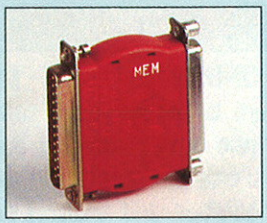
For further information on these or related positions call
JEREMY WILLAN or SARAH HOLTHAM on
(0734) 774234 or (0604) 33195 after 7p.m. FAX: (0734) 772773
Or write in confidence to CPS at:-

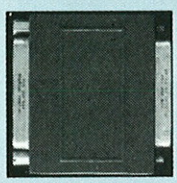
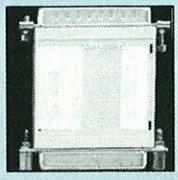
63 Peach Street
Wokingham
Berkshire RG11 1XP

CPS
COMPUTEC PERSONNEL SERVICES

SOFTWARE PROTECTION!

Can You Spot The Difference?

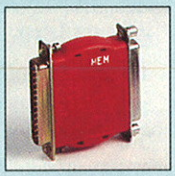



- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 70 languages supported
- ☑ Compatibility - due to 8 years experience & 600,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

OVER 3,100 CUSTOMERS WORLDWIDE HAVE!

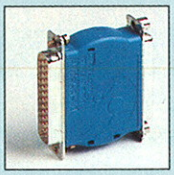
Electronic Key

The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.



1 Word Memory Key

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.

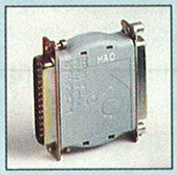


31 Word Memory Key

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.

Micro Processor Key

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.



Macintosh Key

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

MICROPHAR... The European Standard!

For a no-nonsense informative discussion on how our dongles can better protect your profits, please call us on:

091-378 9191

CLEARSOFT

CLEARSOFT, Littleburn Ind. Estate,
Langley Moor, Durham DH7 8HG.
Telephone: (091) 378 9191 Fax: (091) 3789393

Please send me full details on your protection systems.

Name: .....

Position: .....

Address: .....

.....

Telephone: .....

Fax: .....

Signature: .....

Clearsoft, Littleburn Ind Est,
Langley Moor, Durham, DH7 8HG
Tel: 091 3789191 Fax: 091 3789393

.EXEIO/91

CIRCLE NO. 245